

程序员是怎样炼成的？在持续激励中磨砺，在反复练习中成长，PHP疯狂特训内容，尽在本书中……

学通PHP的24堂课

潘凯华　刘欣　杨明　李慧　等编著

130集大型多媒体教学视频

368个中小实例训练　　580余段源码分析
226个应用模块精解　　6大项目案例展示



DVD大型多媒体光盘

130集教学视频：130集（423段）多媒体教学视频，听程序员现场讲解
368个中小实例：夯实必备知识，强化基本功训练
580余源码分析：寻找编程感觉，培养编程思想
226个应用模块：激发学习兴趣，突出开发实战
6大项目案例：体验项目开发过程，积累项目开发经验

清华大学出版社

软件开发羊皮卷

学通 PHP 的 24 堂课

（130 集大型多媒体教学视频）

潘凯华　刘　欣　杨　明　等编著

清华大学出版社

北　京

内 容 简 介

本书以 24 堂课的形式，从初中级用户的角度出发，突出学、练、用相结合，用科学合理的设计全面讲述了使用 PHP 进行程序开发的必备知识和技能，突出学、练、用相结合。主要包括 PHP 概述、PHP 基础、PHP 函数、PHP 流程控制语句、PHP 数组应用、PHP 与 Web 页面交互、MySQL 数据库技术、PHP 数据库编程技术、Cookie 与会话控制、字符串高级处理技术、日期和时间的处理技术、图形图像处理技术、文件目录处理技术、面向对象的程序设计、PHP 与 XML 技术、PDO 数据库抽象层、Smarty 模板技术、Zend Framework 框架、PHP 的安全与防护、PHP 的字符编码、程序调试与错误处理、应用 Smarty 模板开发电子商务网站、Zend Framework 框架开发明日搜索引擎和学校图书馆管理系统。

本书适合有志于从事软件开发的初学者、高校计算机相关专业学生和毕业生，也可作为软件开发人员的参考手册，或者高校的教学参考书。

本书通过教学视频、实例训练、综合应用、项目实践、自我测试、行动指南逐步深入和强化训练等方式，并辅之以心理励志，来持续激发读者主动学习、自发学习。

本书给出了 368 个小型实例，226 个综合应用，6 个项目案例（部分在光盘中），各类技巧、试验 200 余个，测试题目 210 个，以方便读者训练、测试和快速提升。

本书 DVD 光盘给出了 130 集（423 段）多媒体教学视频讲解，每个实例都给出了相应的源程序，可直接复制源码学习或应用。

图书在版编目（CIP）数据

学通 PHP 的 24 堂课/潘凯华，刘欣，杨明等编著．—北京：清华大学出版社，2011.6

（软件开发羊皮卷）

ISBN 978-7-302-25542-0

I. ①学…　II. ①潘…　②刘…　③杨…　III. ①PHP 语言-程序设计　IV. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 091874 号

责任编辑：刘利民
版式设计：文森时代
责任校对：柴　燕
责任印制：
出版发行：清华大学出版社　　　**地　　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　　**邮　　编：**100084
社 总 机：010-62770175　　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：203×260　**印　　张：**46.5　**字　　数：**1496 千字
（附 DVD 光盘 1 张）
版　　次：2011 年 6 月第 1 版　**印　　次：**2011 年 6 月第 1 次印刷
印　　数：1～5000
定　　价：79.80 元

产品编号：042547-01

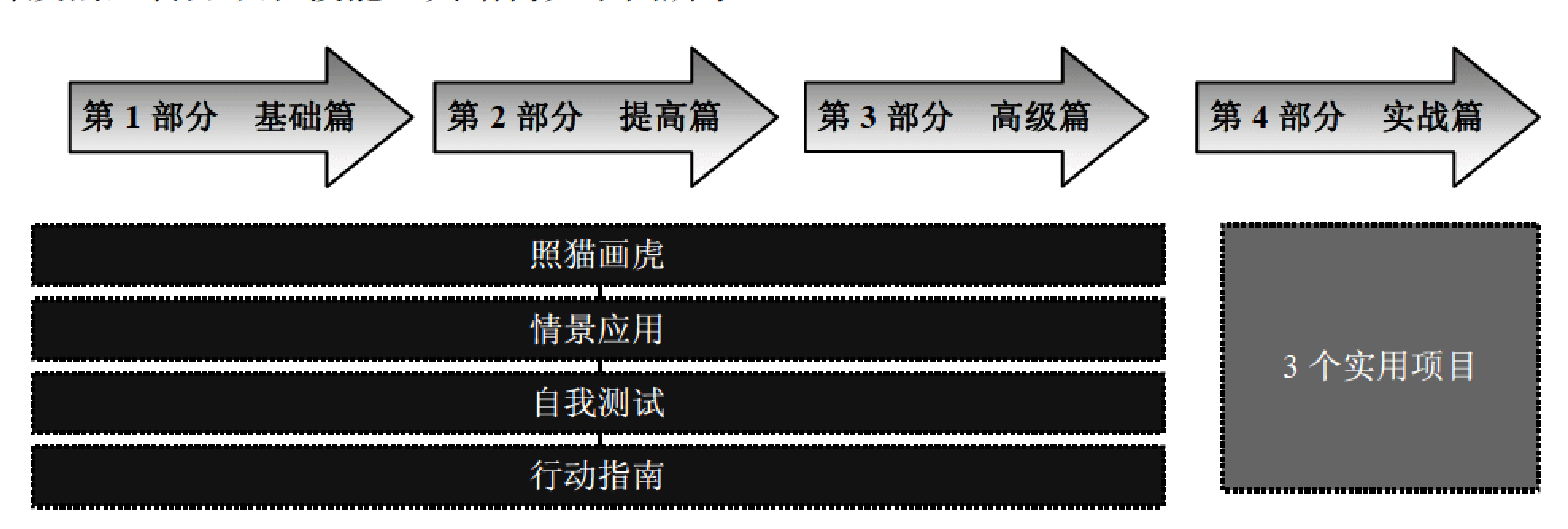
前　言

Preface

PHP 是全球最普及、应用最广泛的互联网开发语言之一。PHP 语言具有简单、易学、源码开放、可操纵多种主流与非主流的数据库、支持面向对象的编程、支持跨平台的操作以及完全免费等特点，越来越受到广大程序员的青睐和认同。PHP 目前拥有几百万名用户，其发展速度要快于它之前的任何一种计算机语言，相信 PHP 在经过不断发展后，一定会成为互联网开发语言中“主流的主流”。

本书内容

学、练、用到精通只需 24 堂课。本书从初中级用户的角度出发，通过 24 堂课全面讲述了使用 PHP 进行程序开发的必备知识和技能。其结构如下图所示。



第 1 部分　基础篇（第 1～8 堂课）： 讲述了 PHP 概述、PHP 基础、PHP 函数、PHP 流程控制语句、PHP 数组应用、PHP 与 Web 页面交互、MySQL 数据库技术和 PHP 数据库编程技术。通过本篇的学习，读者足以搭建起学习 PHP 的框架。

第 2 部分　提高篇（第 9～13 堂课）： 讲述了 Cookie 与会话控制、字符串高级处理技术、日期和时间的处理技术、图形图像处理技术和文件目录处理技术。通过本篇的学习，读者能够掌握 PHP 的常用技术。

第 3 部分　高级篇（第 14～21 堂课）： 讲述了面向对象的程序设计、PHP 与 XML 技术、PDO 数据库抽象层、Smarty 模板技术、Zend Framework 框架、PHP 的安全与防护、PHP 的字符编码和程序调试与错误处理。通过本篇的学习，读者可以学习到 PHP 语言中更高一个层次的技术。

第 4 部分　实战篇（第 22～24 堂课）： 讲述了应用 Smarty 模板开发电子商务网站、Zend Framework 框架开发明日搜索引擎和学校图书馆管理系统 3 个完整的项目实例设计全过程。通过本篇学习，读者可以积累项目开发经验。

本书特点

- **配备 130 集（423 段）多媒体教学视频讲解。**

  本书 DVD 光盘提供了覆盖全书的语音视频讲解，读者可以通过视频快速、直观、轻松地学习。

- ❑ **每一堂课都结合“照猫画虎”、“情景应用”。**
  为了增强读者动手能力，激发学习兴趣，本书提供了“照猫画虎”和“情景应用”栏目，根据实例模仿着去做是学习编程的最快方式。
- ❑ **每一堂课都结合励志故事和“行动指南”，时刻激励和鼓舞读者。**
  我们认为学习中尤其是最初的一段时间，非常有必要不断地对学习者加以激励和鼓舞，让他们坚持下来是至关重要的，因此书中不间断地用一些励志故事和“行动指南”去鼓舞其信心。
- ❑ **光盘提供了本书的所有代码，即使只有一行。**
  本书光盘不仅提供了所有实例的源程序，还提供了书中所有示例的源代码，哪怕只有一行。读者可以直接复制，以提高学习效率。
- ❑ **本书所有习题和实战都给出了答案，部分习题还有解析，读者可以对照查阅。**

读者对象

- ☑ 有志于从事软件开发的初学者
- ☑ 准备从事软件开发的求职者
- ☑ 初、中级程序开发人员
- ☑ 高等院校计算机相关专业的老师和学生
- ☑ 参与毕业设计的学生
- ☑ 程序测试及维护人员

本书作者

本书由明日科技组织编写，参加编写的程序员有杨明、李慧、刘欣、潘凯华、王国辉、赛奎春、陈丹丹、高春艳、李伟、孙秀梅、杨丽、刘玲玲、朱晓、刘燕、陈英、李鑫、李贺、肖鑫、张丽娜、沈博、刘冠男、曹飞飞、李丽、聂喜婷、王明昭、张英豪、王小科、赵会东、白伟明等。

由于水平有限，书中疏漏和不足之处在所难免，恳请广大读者朋友批评指正。

技术支持与服务

秉持“十年服务，始终如一”的理念，我们承诺如果您在学习或使用本书的过程中遇到问题或疑惑，可以通过如下方式与我们联系。

- ❑ 登录技术服务网站：www.mingribook.com，查阅相关问题或者留言。
- ❑ 通过企业服务邮箱：tmoonbook@sina.com 或 th_press@263.net。
- ❑ 申请加入服务 QQ：100310063。

我们承诺将在 5 个工作日内给您提供解答。

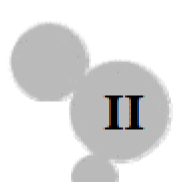
最后，感谢您选择本书，希望本书能成为您编程路上的领航者。

祝读书快乐！

特别提醒：

亲爱的读者朋友，由于近期纸张价格和印制成本大幅上涨，为不增加读者朋友的负担，又不减少书的内容，本书的最后几章内容不得不放在配书光盘中（参见目录），由此给您带来了不便，在此深表歉意。

编　　者

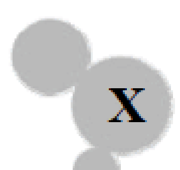


目　录

Contents

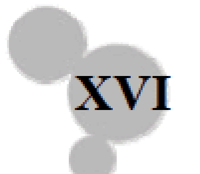
第 1 部分　基础篇

第 2 部分 提高篇



第 3 部分　高级篇

第 4 部分 实战篇



特别提醒：

亲爱的读者朋友，由于近期纸张价格和印制成本大幅上涨，为不增加读者朋友的负担，又不减少书的内容，本书以下章节内容不得不放在配书光盘中，由此给您带来了不便，在此深表歉意。



基础篇

第 1 堂课

PHP 概述

（视频讲解：110 分钟）

PHP 是一种服务器端的嵌入式脚本语言，是一种服务器端、跨平台、面向对象、HTML 嵌入式的脚本语言。本堂课将向读者简单介绍 PHP 语言、PHP 的语言优势、下载 PHP 及相关软件、搭建 PHP 的开发环境，了解常用的配置信息，熟悉 PHP 开发环境的配置结构。其主要目的是让读者先在宏观上对 PHP 语言有一个整体的了解，使读者对 PHP 有一个总体的认识，并找到学习 PHP 的切入点。

学习摘要：

- 怎样学好 PHP
- PHP 环境的搭建
- PHP 开发环境的关键配置信息
- 解决 PHP 的常见配置问题
- 通过实例了解 PHP 配置环境的相关信息

1.1 怎样学好 PHP

1.1.1 什么是 PHP

PHP 是 Hypertext Preprocessor（超文本预处理器）的缩写，是一种服务器端、跨平台、HTML 嵌入式的脚本语言。其独特的语法混合了 C 语言、Java 语言和 Perl 语言的特点，是一种被广泛应用的、开源式的多用途脚本语言，尤其适合 Web 开发。

1.1.2 PHP 的优势

PHP 起源于 1995 年，由 Rasmus Lerdorf 开发。目前已有超过 2200 万个网站、1.5 万家公司、450 万程序开发人员在使用 PHP 语言，它是目前动态网页开发中使用最为广泛的语言之一。PHP 是生于网络、用于网络、发展于网络的一门语言，它一诞生就被打上了自由发展的烙印。目前在国内外有数以千计的个人和组织的网站在以各种形式和各种语言学习、发展和完善它，并不断地公布最新的应用和研究成果。PHP 能运行在包括 Windows、Linux 等在内的绝大多数操作系统环境中，常与免费 Web 服务器软件 Apache 和免费数据库 MySQL 配合使用于 Linux 平台上，具有很高的性价比，这 3 种技术的结合号称“黄金组合”。下面介绍 PHP 开发语言的特点。

☑ 速度快

PHP 是一种强大的 CGI 脚本语言，混合了 C、Java、Perl 和 PHP 式的新语法，执行网页速度比 CGI、Perl 和 ASP 更快，而且内嵌 Zend 加速引擎，性能稳定。这是它的第一个突出特点。

☑ 支持面向对象

面向对象编程（OOP）是当前的软件开发趋势，PHP 对 OOP 提供了良好的支持。可以使用 OOP 的思想来进行 PHP 的高级编程，对于提高 PHP 编程能力和规划好 Web 开发构架非常有意义。

☑ 实用性强

由于 PHP 是一种面向对象的、完全跨平台的新型 Web 开发语言，所以无论从开发者角度考虑还是从经济角度考虑，都是非常实用的。PHP 语法结构简单，易于入门，很多功能只需一个函数就可以实现，并且很多机构都相继推出了用于开发 PHP 的 IDE 工具。

在 TIOBE 网站发布的最新《2010 年 11 月份编程语言排名》中，Java 仍然跃居榜首，PHP 编程语言的增长势头与 C++十分接近，专家分析 PHP 有望超过 C++跃居排行榜三甲，如图 1.1 所示。

2010年9月排行榜

2010年9月排行	2009年9月排行	程序设计语言	2010年9月所占比率
1	1	Java	17.915%
2	2	C	17.147%
3	4	C++	9.812%
4	3	PHP	8.370%
5	5	(Visual)Basic	5.797%
6	7	C#	5.016%
7	8	Python	4.583%
8	18	Objective-C	3.368%
9	6	Perl	2.447%
10	10	Ruby	1.907%
11	9	JavaScript	1.665%
12	11	Delphi	1.585%
13	13	Lisp	1.084%
14	12	Pascal	0.790%

图 1.1 在 TIOBE 网站发布的最新《2010 年 11 月份编程语言排名》

说明： TIOBE 编程语言排行榜衡量了各种编程语言的流行程度。该排行榜每月发布一次，统计数据包括全球范围的软件工程师、培训课程以及第三方供应商，数据来自 Google、MSN 和 Yahoo!等流行搜索引擎。

近几年 PHP 呈现上升趋势，如图 1.2 所示，这也正说明 PHP 语言简单、易学、面向对象和安全等特点



正在被更多人所认同。相信新的 PHP 语言将会朝着更加企业化的方向迈进，并且将更适合大型系统的开发。

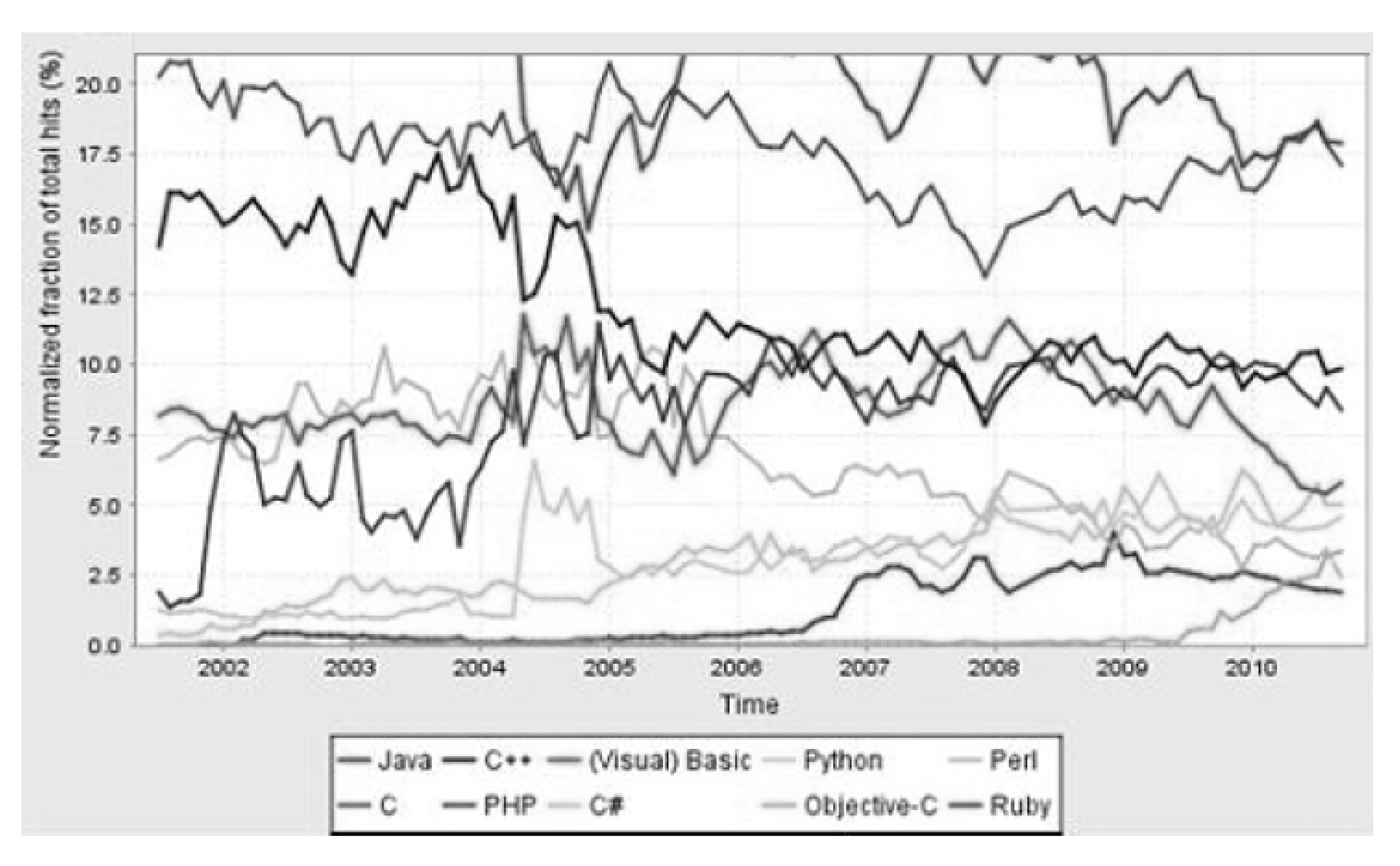


图 1.2　PHP 的发展趋势

说明：图 1.1、图 1.2 中的数据摘自 http://www.tiobe.com 网站。

☑　功能强大

PHP 在 Web 项目开发过程中具有极其强大的功能，而且实现相对简单，主要表现在如下几点：

➢　可操纵多种主流与非主流的数据库，如 MySQL、Access、SQL Server、Oracle、DB2 等，其中，PHP 与 MySQL 是现在绝佳的组合，可以跨平台运行。
➢　可与轻量级目录访问协议进行信息交换。
➢　可与多种协议进行通信，包括 IMAP、POP3、SMTP、SOAP 和 DNS 等。
➢　使用基于 POSIX 和 Perl 的正则表达式库解析复杂字符串。
➢　可以实现对 XML 文档进行有效管理及创建和调用 Web 服务等操作。

☑　可选择性

PHP 可以采用面向过程和面向对象两种开发模式，并向下兼容，开发人员可以从所开发网站的规模和日后维护等多角度考虑，以选择所开发网站应采取的模式。

PHP 进行 Web 开发过程中使用最多的是 MySQL 数据库。PHP 5.0 以上版本中不仅提供了早期 MySQL 数据库操纵函数，而且提供了 MySQLi 扩展技术对 MySQL 数据库的操纵，这样开发人员可以从稳定性和执行效率等方面考虑操纵 MySQL 数据库的方式。

☑　成本低

PHP 具有很好的开放性和可扩展性，属于自由软件，其源代码完全公开，任何程序员为 PHP 扩展附加功能非常容易。在很多网站上都可以下载到最新版本的 PHP。目前，PHP 主要是基于 Web 服务器运行的，支持 PHP 脚本运行的服务器有多种，其中最有代表性的为 Apache 和 IIS，PHP 不受平台束缚，可以在 UNIX、Linux 等众多版本的操作系统中架设基于 PHP 的 Web 服务器。采用 Linux+Apache+PHP+MySQL 这种开源免费的框架结构可以为网站经营者节省很大一笔开支。

☑　版本更新速度快

与数年才更新一次的 ASP 相比，PHP 的更新速度要快得多，因为 PHP 几乎每年更新一次。

☑　模板化

实现程序逻辑与用户界面分离。

☑ 应用范围广

目前在互联网上有很多网站的开发都是通过 PHP 语言来完成的，如搜狐、网易和百度等，在这些知名网站的创作开发中都应用到了 PHP 语言。

1.1.3 如何学好编程

俗语说：“水滴石穿，绳锯木断”。无论做什么事情，都必须坚定信心，保持高昂的斗志，持之以恒是做事的根本。学习 PHP 同样是如此，首先必须坚定信心，让自己充满激情，然后脚踏实地，认认真真地去学习，去迎接每一个挑战，只有经过暴风雨的洗礼后，才能看到绚丽的彩虹。程序员的成长之路如图 1.3 所示。

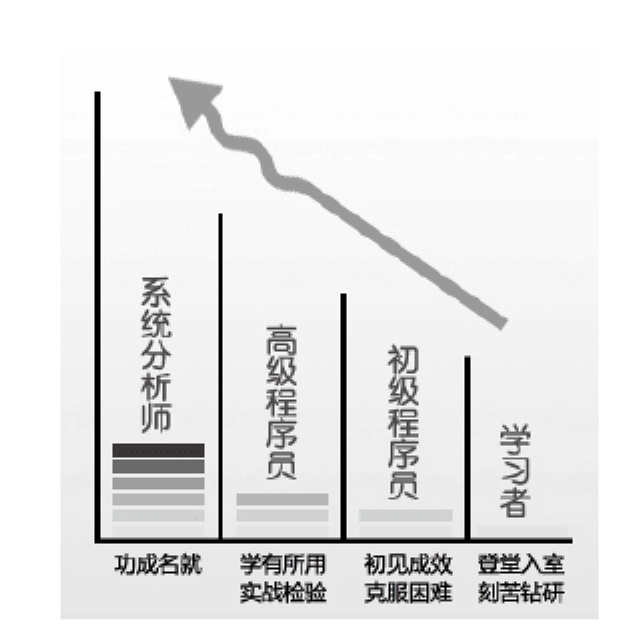


图 1.3 程序员的成长之路

学习 PHP 语言，应该讲究方法、策略，别人的学习经验可以借鉴，但不要照抄照搬，因为有些东西也许并不适合你自己。应该学会自己总结、分析，整理出一套适合自己的学习方法。

下面是笔者结合多年的开发总结出来的学习经验与所有 PHP 程序开发者分享。切忌“照抄照搬”，因为有些东西也许并不适合你自己。

（1）熟悉 HTML/CSS/JavaScript 等网页基本元素，完成阶段可自行制作完整的网页，以对元素属性达到熟悉程度。

任何网站都是由网页组成的，也就是说想完成网站，必须先学会做网页，因此必须要掌握 HTML 语言，才能为今后制作网站打下基础。掌握静态网页的制作技术是学习开发网站的先决条件。

在学习 HTML、CSS 和 JavaScript 的过程中，边学边做是最有效的方式，当然这一方式对于学习 PHP 同样是最有效的。HTML 中的任何元素都要亲自实践，只有熟悉各个元素会起到什么效果之后，才会记忆深刻，而对于书本中的知识只是一味地死记硬背是绝对不行的，大部分新手之所以觉得概念难学，是因为“懒惰”，没有通过实际的操作来理解概念的意义。

（2）学会配置 PHP 的开发环境，并选择一种适合自己的开发工具。

搭建 PHP 环境，可以选择独立安装，即自行配置 Apache 服务器、MySQL 服务器和 PHP；也可以选择集成化安装包。对于新手来说，建议使用集成化安装包，因为集成化安装包操作起来非常方便，可以直接安装 Apache 服务器、MySQL 服务器、PHP 和 phpMyAdmin，而且不需要用户手动配置，安装完以后就可以直接使用。例如，使用 AppServ 集成化安装包（在 1.2.1 节有详细介绍）。

开发工具的选择，应该选择适合自己的，而不要追求流行的。PHP 的编辑工具有很多，每款编辑工具也各有各的优势。在编写程序时，一款好的编辑工具会使程序员编写过程更加轻松、有效和快捷，达到事半功倍的效果。在 1.1.5 节中推荐了几款开发工具，读者可以结合自己的需求进行选择。

（3）理解动态语言的概念和运作机制，熟悉 PHP 语法。

（4）学习如何将 PHP 与 HTML 结合起来完成简单动态页面。

一个简单漂亮的 PHP 小实例的开发，意味着你在 PHP 的编程之路上已经迈出了一大步。

（5）接触 MySQL 数据库，开始设计数据库方面的程序。

MySQL 数据库是 PHP 语言的最佳搭档，有关它的知识是必须掌握的，包括数据库的连接、查询、添加、修改和删除等。

（6）不断巩固和熟练使用大部分 PHP 常用函数，并且理解 OOP、MySQL 优化，以及各种模板（如 Smarty、ADODB）。



（7）掌握 PHP 的编码规范，以此来养成一个良好的编程习惯。

（8）手中常备两册一书。

手中常备 PHP 中文手册和 MySQL 手册，以及一本 PHP 基础类书籍。它们可以帮助你解决程序开发中遇到的一些问题，同时也能够提高编程效率。

（9）理论联系实际，坚定信念。

勤动手，多向他人请教，学习他人的编程思想，学会去其糟粕，取其精华；要有愚公移山、铁杵磨成针的精神；不抛弃，不放弃。

1.1.4　下载 PHP 及相关软件

搭建 PHP 环境涉及系统平台、Web 服务软件和数据库软件及 PHP 本身。根据自身现有计算机软、硬件环境，可以自由选择相应的软件。

通常选择 Windows NT 为实验平台，这样可以下载 PHP 5.0 以上的 Windows 版本。数据库可以下载 MySQL 的 Windows 版本（www.MySQL.org）或者使用微软公司的 MsSQL。Web 服务软件可以直接下载 Apache 的 Windows 版本（www.apache.com）。

如果想搭建 Linux 下运行的实战环境，那么所有这些软件必须下载其对应于 Linux 的版本，有的可能需要在 Linux 下编译生成。

下面以 Windows 版本为例，简单说明一下 PHP 优秀的集成开发环境及相关信息。

（1）XAMPP 是一个易于安装且包含 MySQL、PHP 和 Perl 的 Apache 发行版。只需根据提示，即可安装成功。不必对 PHP、Apache、MySQL 配置文件进行修改及相关繁琐的操作（例如，将 PHP 的配置文件 php.ini 保存到操作系统 C 盘下的 WINDOWS 文件夹下，手动开启 MySQL 组件、Oracle 组件和 GD2 支持等），大大省去了初学者在配置运行环境时的时间，真正意义上做到了一键安装，开发运行的理念。官方下载地址如图 1.4 所示。

（2）AppServ 将 Apache、PHP、MySQL 和 phpMyAdmin 等服务器软件和工具安装配置完成后打包处理，同 XAMPP 一样安装相当简单。官方下载地址如图 1.5 所示。

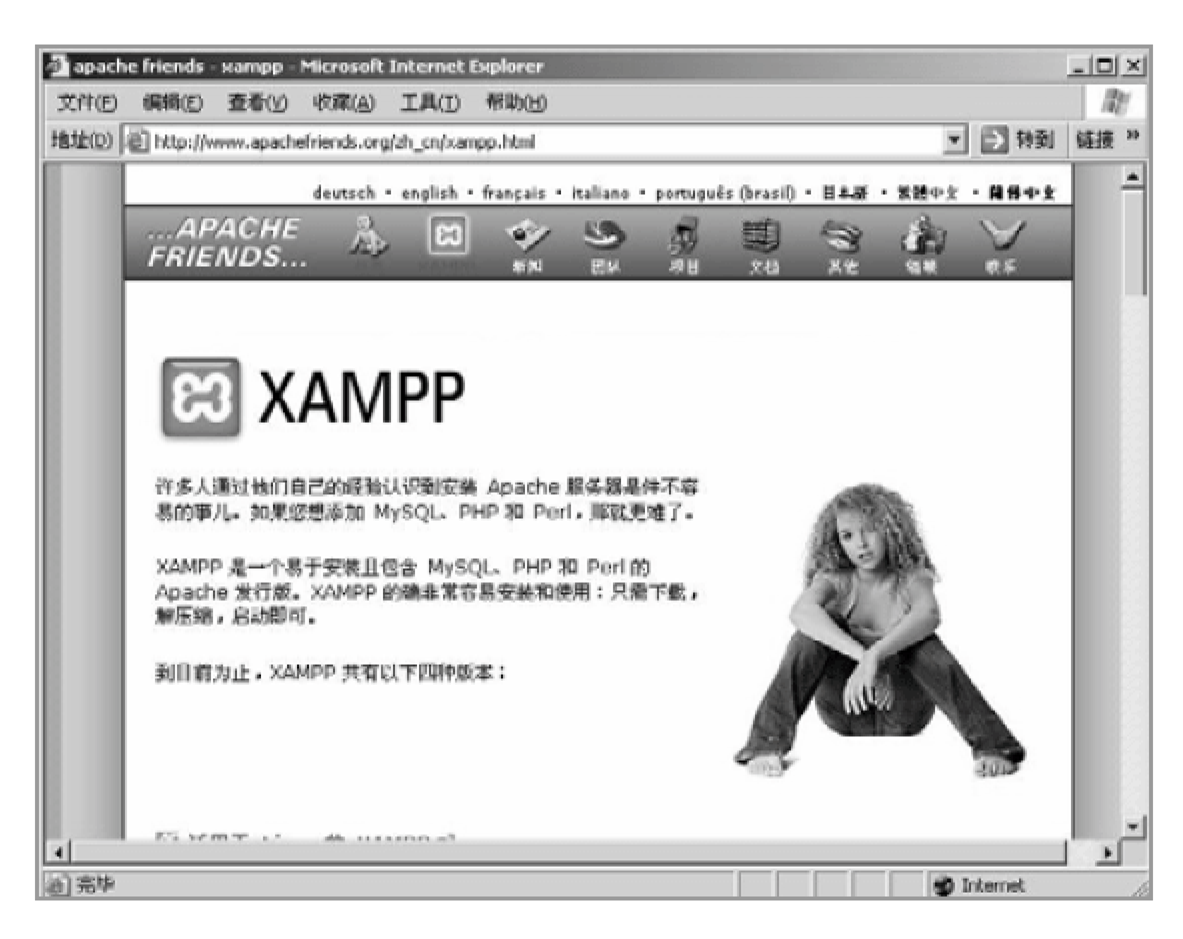


图 1.4　官方下载地址（XAMPP）



图 1.5　官方下载地址（AppServ）



1.1.5　代码编辑工具

选择 PHP 的代码编辑工具，应该考虑以下 4 个方面的因素：

（1）语法的高亮显示。应用语法的高亮显示，可以对代码中的不同元素采用不同的颜色进行显示，如关键字用蓝色、对象方法用红色标识等。

（2）格式排版功能。格式排版功能可以使程序代码的组织结构清晰易懂，并且易于程序员进行程序调试，排除程序的错误异常。

（3）代码提示功能。代码提示功能可以在程序员编写某个函数时，提供这个函数的语法信息，甚至可以在程序员输入某个字符时，给出这个字符相关的函数信息，可以帮助程序员编写正确的函数，使用正确的语法。

（4）界面设计功能。不但可以编写 PHP 代码，还可以进行界面的设计。

这是在选择代码编辑工具时应该考虑的问题，这 4 个因素不可能都完全满足，应该根据自己的实际情况进行选择。

下面介绍几款常用代码编辑工具，供广大读者参考。

☑　Macromedia Dreamweaver

Macromedia Dreamweaver 是一款专业的网站开发编辑器。它将可视布局工具、应用程序开发功能和代码编辑支持组合在一起，其功能强大，使得各个层次的开发人员和设计人员都能够快速创建出吸引人的、标准的网站和应用程序。它采用了多种先进的技术，能够快速高效地创建极具表现力和动感效果的网页，使网页创作过程简单无比。同时，Macromedia Dreamweaver 提供了代码自动完成功能，不但可以提高编写速度，而且减少了错误代码出现的几率。Macromedia Dreamweaver 既适用于初学者制作简单的网页，又适用于网站设计师、网站程序员开发各类大型应用程序，极大地方便了程序员对网站的开发与维护。

Macromedia Dreamweaver 从 MX 版本开始支持 PHP+MySQL 的可视化开发，对于初学者确实是比较好的选择，因为如果是一般性开发，几乎是不写一行代码也可以写出一个程序，而且都是所见即所得的。所包含的特征包括语法加亮、函数补全、形参提示、全局查找替换、处理 Flash 和图像编辑等。同时，为 PHP、ASP 等脚本语言提供辅助支持。

下载地址为 http://www.adobe.com/downloads/。

☑　ZendStudio

ZendStudio 是目前公认的最强大的 PHP 开发工具，但也是一款收费软件。其具备功能强大的专业编辑工具和调试工具，包括了编辑和调试，配置 PHP 程序所需要的客户及服务器组件，支持 PHP 语法加亮显示，尤其是功能齐全的调试功能，能够帮助程序员解决在开发中遇到的问题。

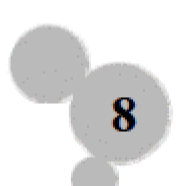
下载地址为 http://www.zend.com/store/products/zend-studio.php。

📖 **说明：**ZendStudio 虽然是收费软件，但如果是个人使用，可以下载试用版。

☑　PHPEdit

PHPEdit 是一款 Windows 下优秀的 PHP 脚本 IDE（集成开发环境）。该软件为快速、便捷地开发 PHP 脚本提供了多种工具，其功能包括语法关键词高亮，代码提示、浏览，集成 PHP 调试工具，帮助生成器，自定义快捷方式，150 多个脚本命令，键盘模板，报告生成器，快速标记，插件等。

官方网站为 http://phpedit.svoi.net。



1.1.6　下载 PHP 用户手册

获取 PHP 的帮助信息，除了可以到书店购买 PHP 的相关教材外，还可以直接到 PHP 的官方网站中下载 PHP 中文手册，PHP 机构的官方网址为 http://www.php.net。初学者可以在手册中查找到相关函数的详细说明。由于 PHP 代码公开，而且完全免费，所以可以直接在 PHP 的官方网站中下载到 PHP 用户手册。

用户手册是学习 PHP 的良师，许多 PHP 的高手仍然随时备查。手册通常为 PHP 5.0，有中、英文版本，中文版中还有 HTML、ZIP 和 CHM 3 种格式。很多网站都有下载。建议下载 CHM 格式，查阅较为方便。

在学习的过程中可以通过下载 PHP 中文手册和 MySQL 中文手册来获得帮助。为了便于学习，下面推荐两个网址供参考。

（1）PHP 中文手册下载地址：http://www.php.net。PHP 中文手册的运行效果如图 1.6 所示。

（2）MySQL 中文手册下载地址：http://www.mysql.com。MySQL 中文手册的运行效果如图 1.7 所示。

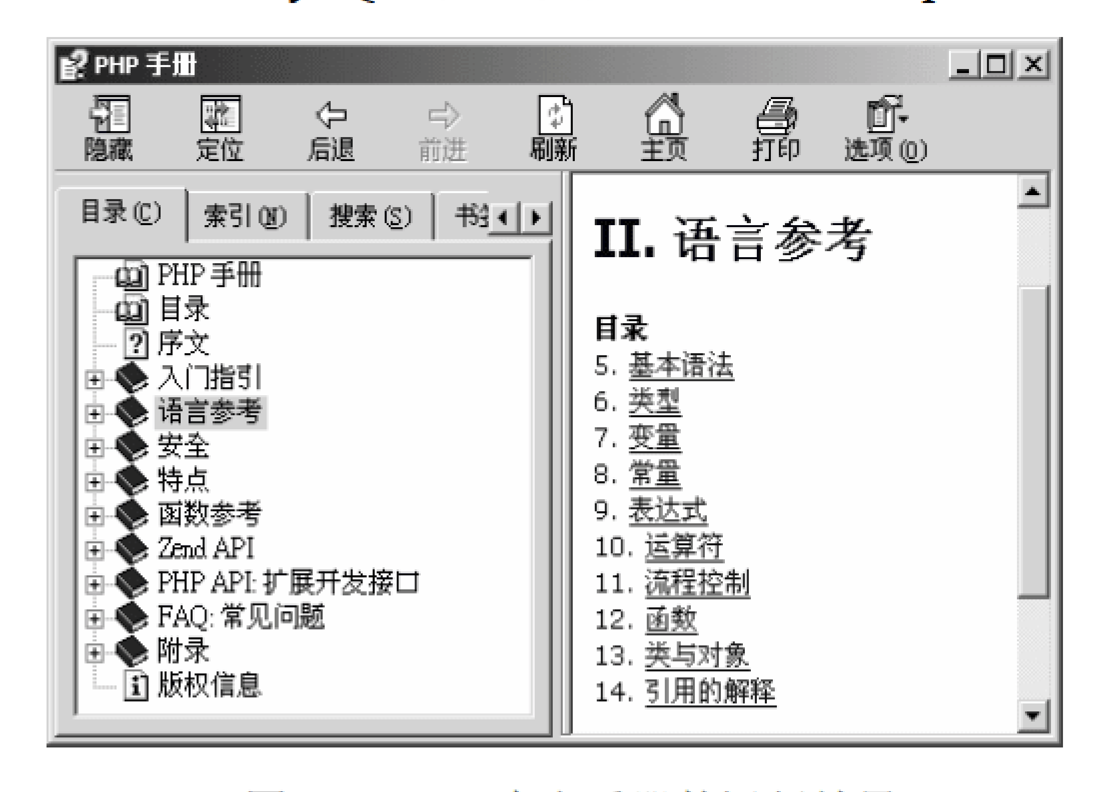


图 1.6　PHP 中文手册的运行效果

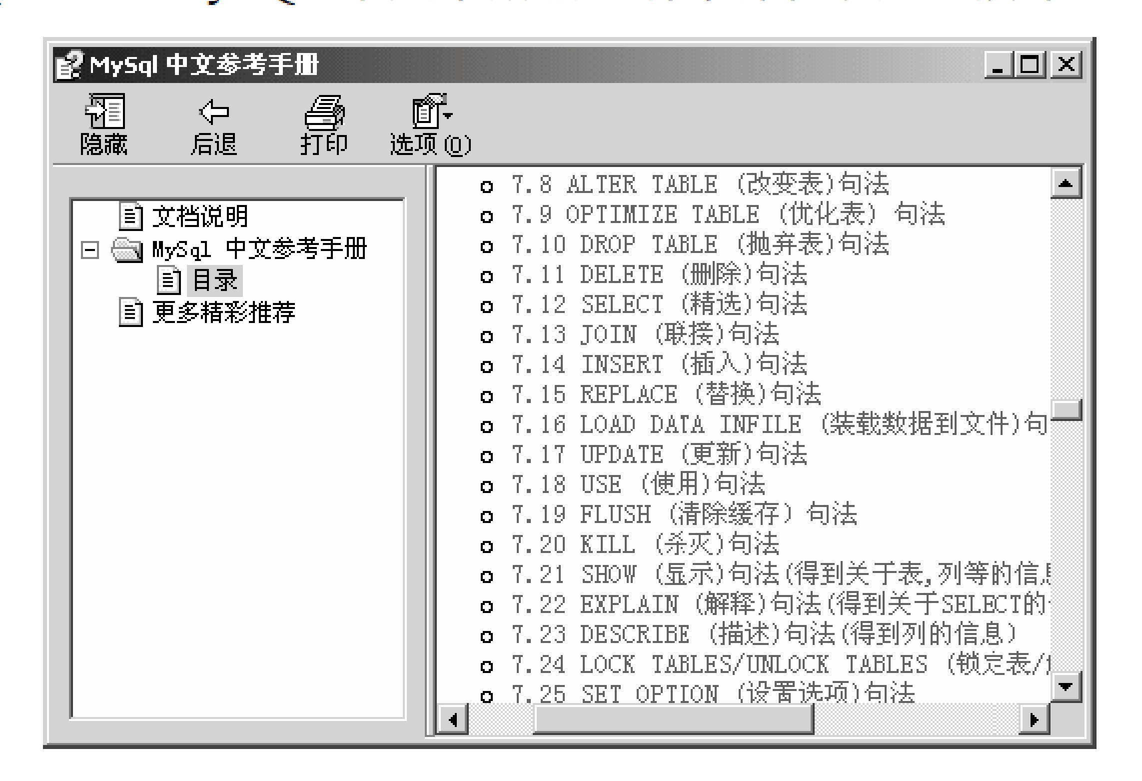


图 1.7　MySQL 中文手册的运行效果

1.2　环境的搭建

1.2.1　AppServ——Windows 版 PHP 集成化安装包

AppServ 将 Apache、PHP、MySQL 和 phpMyAdmin 等服务器软件和工具安装配置完成后打包处理。开发人员只要到网站上下载该软件，然后安装，即可完成 PHP 开发环境的快速搭建。非常适合初学者使用。

注意： 在使用 AppServ 搭建 PHP 开发环境时，必须确保在系统中没有安装 Apache、PHP 和 MySQL。否则，要先将这些软件卸载，然后应用 AppServ。

你问我答： 什么是 PHP 开发环境集成安装包？

目前网络上流行的集成安装包有十几种，其中比较常用是 EasyPHP、AppServ 和 XAMPP，它们都可以搭建 PHP 的开发环境。但是，根据实际的应用情况，还是 AppServ 相对比较好用一些，因为诸如 EasyPHP 或者 XAMPP，在对一些类库的支持上会产生一些错误，导致类库不能正常使用。例如，在 EasyPHP 或者 XAMPP 中应用 jpgraph 类库开发图像时，就会出错。

下面讲解 AppServ 集成化安装包搭建 PHP 开发环境的具体操作步骤。

（1）双击 AppServ-win32-2.5.10.exe 文件，打开如图 1.8 所示的 AppServ 启动页面。

（2）单击图 1.8 中的 Next 按钮，打开如图 1.9 所示的 AppServ 安装协议页面。

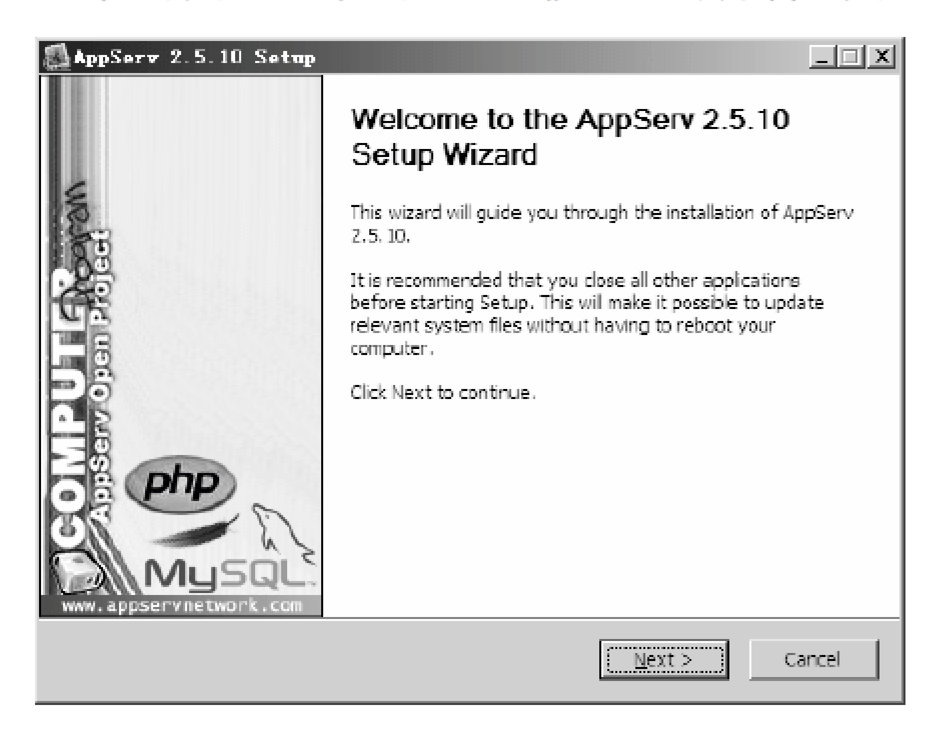


图 1.8　AppServ 启动页面

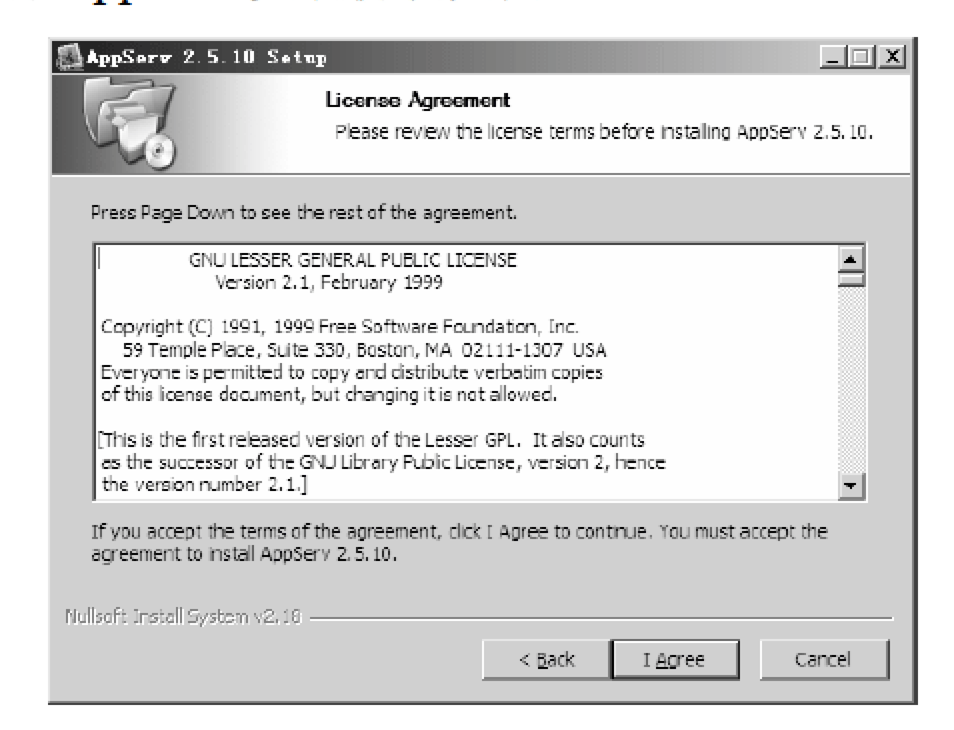


图 1.9　AppServ 安装协议页面

（3）单击图 1.9 中的 I Agree 按钮，打开如图 1.10 所示的页面。设置 AppServ 的安装路径（默认安装路径一般为 E:\AppServ），AppServ 安装完成后，Apache、MySQL、PHP 都将以子目录的形式存储到该目录下。

（4）单击图 1.10 中的 Next 按钮，打开如图 1.11 所示的页面。选择要安装的程序和组件（默认为全选）。

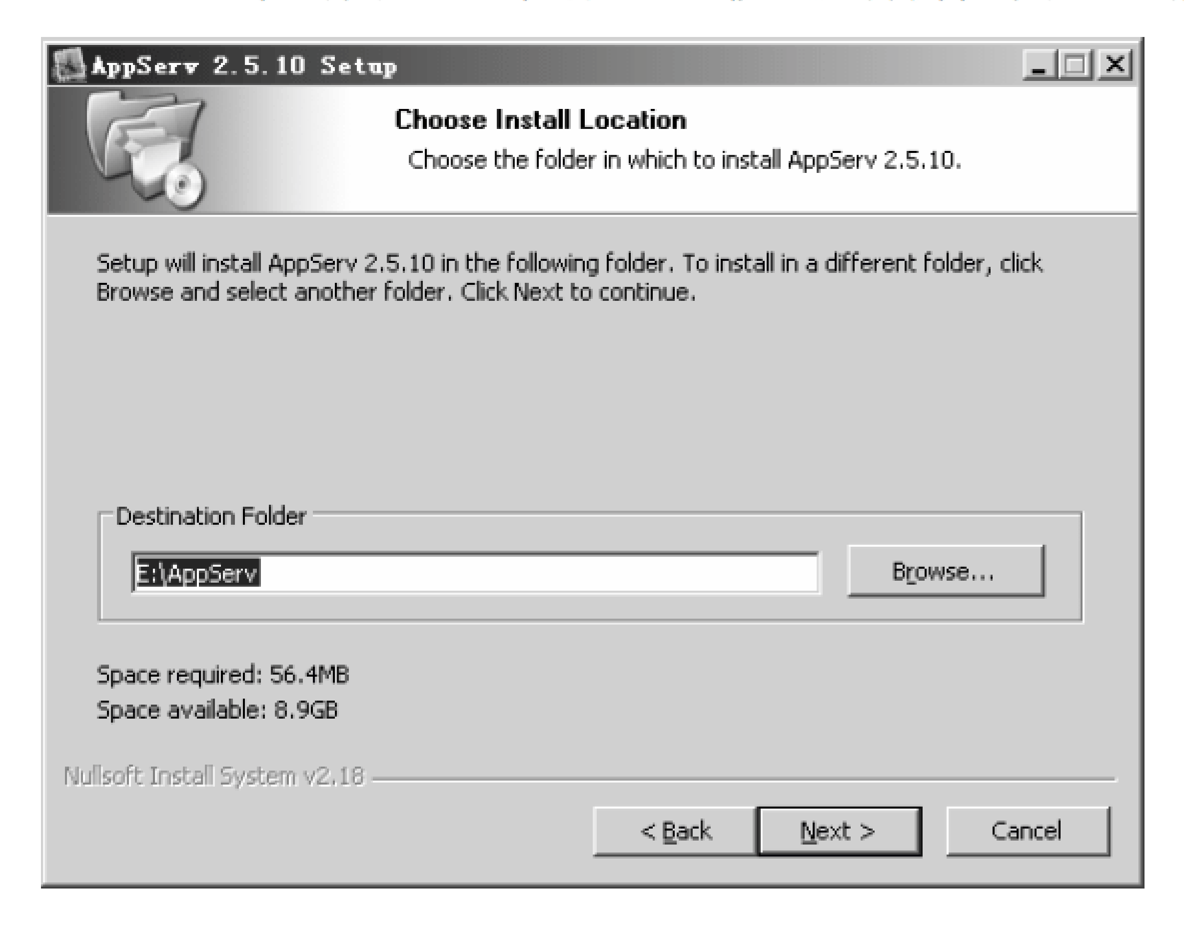


图 1.10　AppServ 安装路径选择

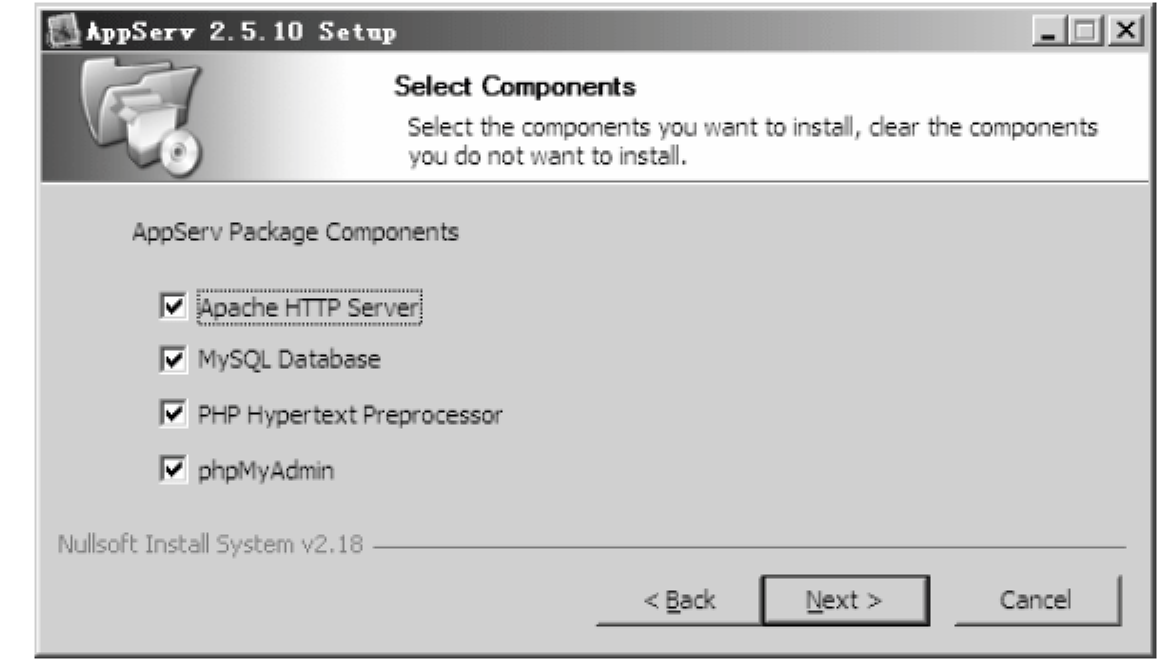


图 1.11　AppServ 安装选项

技巧： 在图 1.11 的操作步骤中，如果本机中已经安装 MySQL 数据库，那么在这里可以不选中 MySQL Database 复选框，仍使用本机已经存在的 MySQL 数据库。

（5）在图 1.11 中单击 Next 按钮，打开如图 1.12 所示的页面。填写计算机名称，添加邮箱地址，设置 Apache 的端口号，默认为 80 端口。

注意： Apache 服务器端口号的设置，直接关系到 Apache 服务器是否能够正常启动。如果本机中的 80 端口被 IIS 或者迅雷占用，那么这里就需要修改 Apache 的端口号，或者将 IIS、迅雷的端口号修改，才能完成 Apache 服务器的配置。如果出现端口冲突，那么将导致安装失败，Apache 服务不能启动。



（6）单击图 1.12 中的 Next 按钮，打开如图 1.13 所示的页面。设置 MySQL 数据库 root 用户的登录密码及字符集。

说明：MySQL 数据库字符集的设置，可以选择 UTF-8、GBK 或者 GB2312。这里将字符集设置为 UTF-8 Unicode，表示 MySQL 数据库的字符集将采用 UTF8 编码。

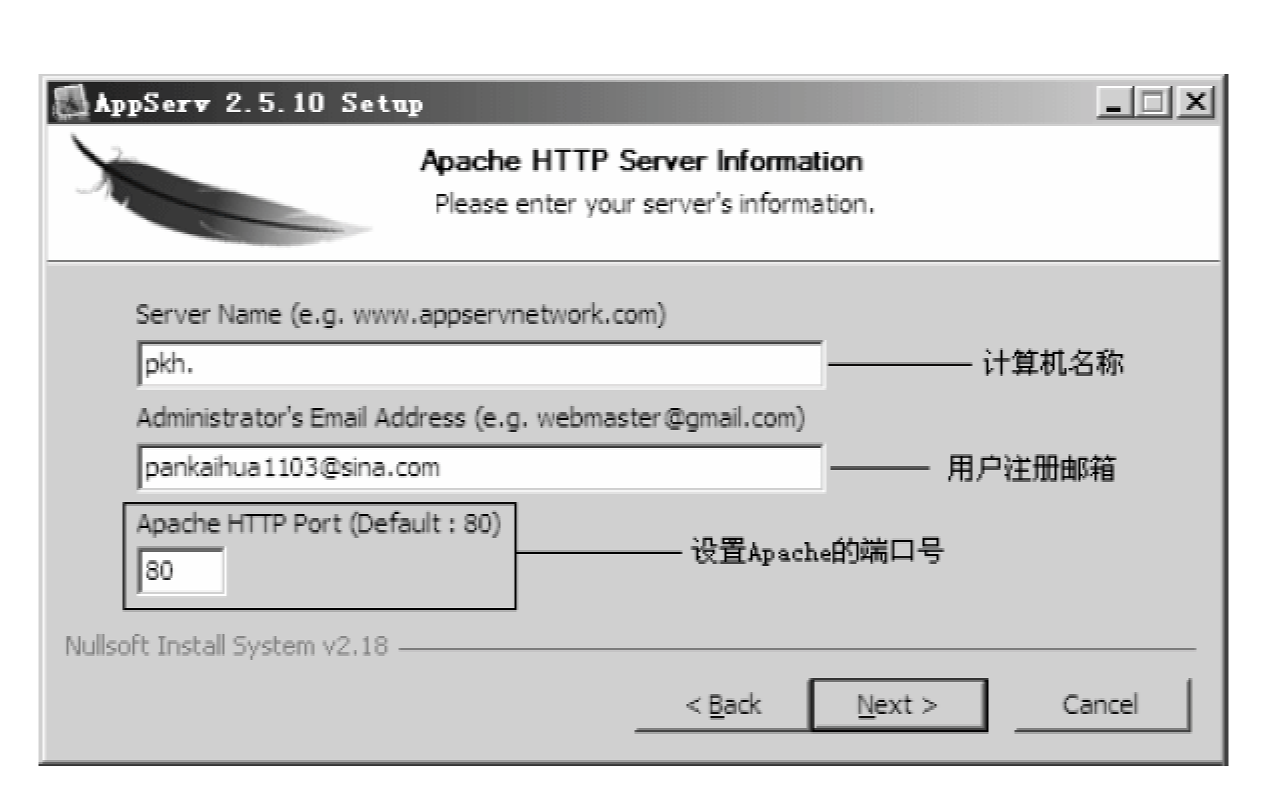


图 1.12　Apache 端口号设置

图 1.13　MySQL 设置

技巧：对于在图 1.13 中设置的 MySQL 数据库 root 用户的密码必须牢记，因为程序在连接数据库时必须使用这个密码。建议读者将这个密码设置为 111。因为这是在开发本书中的程序时所使用的数据库密码。如此，您在运行本书中的数据库程序时，就不需要修改密码；否则，您在运行本书的程序时，要修改连接数据库的密码。如果忘记安装时设置的密码，最直接有效的解决方式是重新安装 AppServ。

（7）单击图 1.13 中的 Install 按钮开始安装，如图 1.14 所示。

（8）安装完成后可以在开始菜单的 AppServ 相关操作列表中启动 Apache 及 MySQL 服务，如图 1.15 所示。

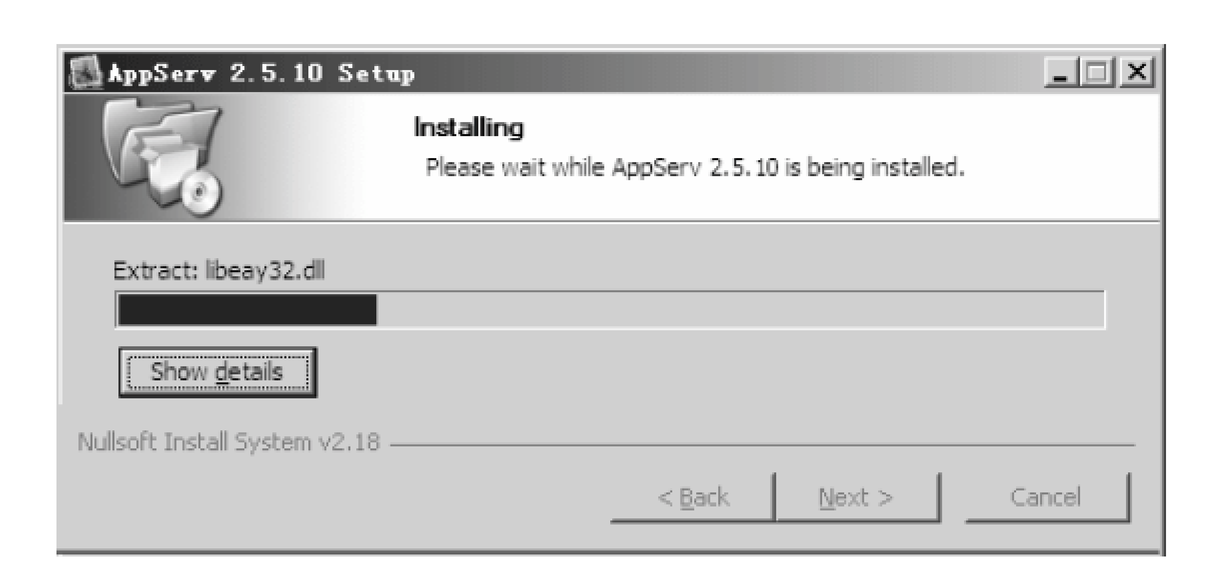


图 1.14　AppServ 安装页面

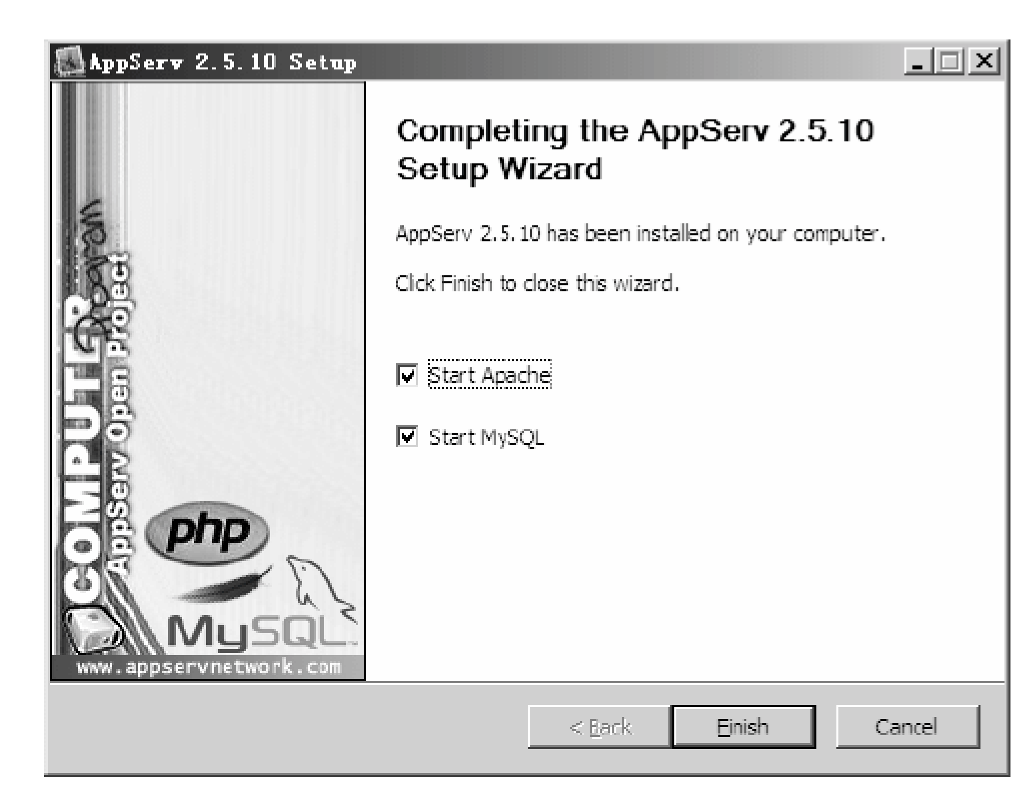


图 1.15　AppServ 安装完成页面

1.2.2　XAMPP——Linux 版 PHP 集成化安装包

本小节介绍在 Linux 操作系统下通过 XAMPP 配置 PHP 的开发环境，以及它的基本应用。XAMPP

（Apache+MySQL+PHP+Perl）是一个功能强大的建站集成软件包。它可以在 Windows、Linux 和 Solaris 3 种操作系统下安装使用，支持多语言，包括英文、简体中文、繁体中文、韩文、俄文和日文等。

在 Linux 操作系统下，使用集成软件 XAMPP 的 Linux 版来配置 PHP 开发环境。Linux 下安装 XAMPP 的步骤如下：

（1）在 Linux 操作系统下，选择“主菜单”/“系统工具”/“终端”命令。

（2）在命令模式下，首先进入到系统的根目录下。

（3）通过 mkdir 命令在根目录下创建一个 opt 目录。

（4）通过 tar xvfz 命令将 XAMPP 解压缩到 opt 目录下。

> **注意：**在 Linux 操作系统下执行 XAMPP 的解压缩时，必须进入到系统的根目录下，并且要在根目录下创建一个子目录，然后才可以执行解压缩的命令；否则，将提示找不到指定的目录或者文件。

（5）按 Enter 键，执行 XAMPP 的解压缩，直到安装成功。其具体使用的命令如图 1.16 所示。

（6）安装成功后，查看/opt/lampp 目录，如图 1.17 所示。

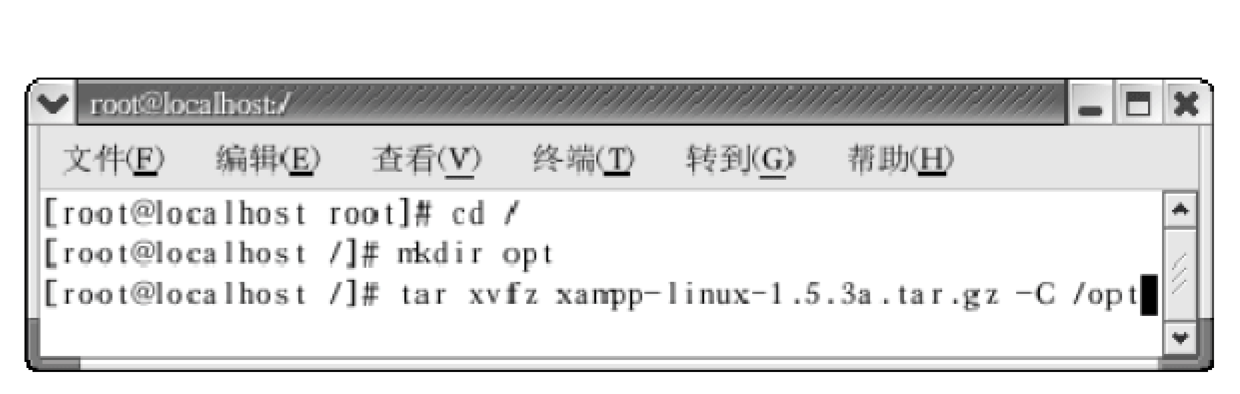


图 1.16　XAMPP 的安装



图 1.17　查看安装的内容

（7）在 Mozilla 浏览器中输入“http://127.0.0.1/xampp/index.php”，运行结果如图 1.18 所示。

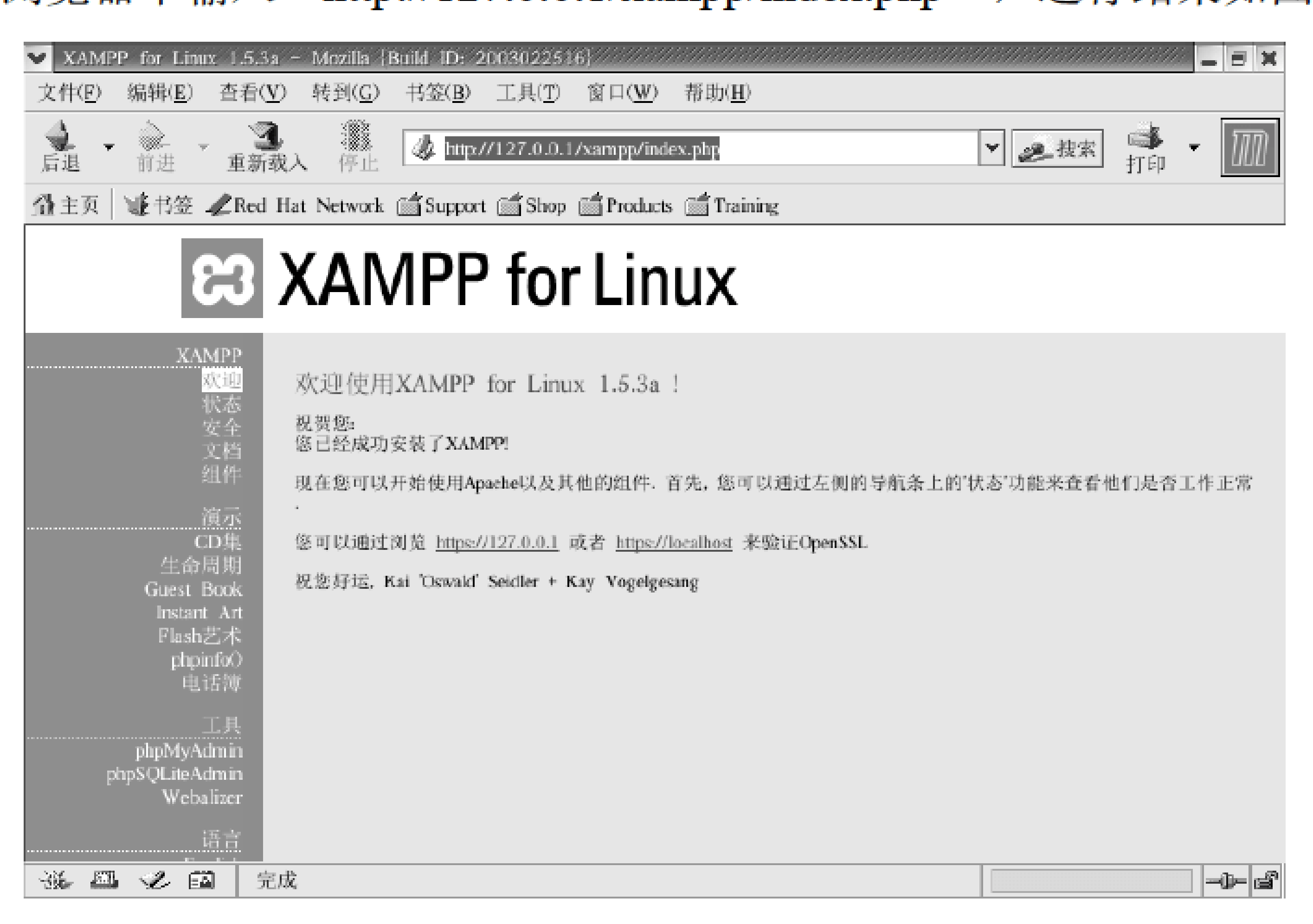


图 1.18　Linux 版的 XAMPP

> **注意：**在 Linux 操作系统下安装 XAMPP，必须在 Linux“主菜单”/“系统工具”下的“终端”命令中通过指令来完成，切忌不要使用任何微软操作系统下的工具来进行操作。

> **技巧：**Linux 下卸载 XAMPP 的命令：rm　rf /opt/lampp。



1.3　PHP 开发环境的关键配置信息

前面介绍了 PHP 开发环境的配置方法，除了安装步骤本身之外，PHP 与服务器的配置也是十分重要的。下面将主要介绍 PHP 及 Apache 服务器的配置。

1.3.1　Apache 服务器的基本配置

Apache 服务器的设置文件在 Linux 操作系统中位于/usr/local/apache/conf/（在 Windows 操作系统中位于“/etc/httpd/conf”）目录下，基本上使用以下 3 个配置文件来配置 Apache 服务器的行为。

☑　access.conf：用于配置服务器的访问权限，控制不同用户和计算机的访问限制。

☑　httpd.conf：用于设置服务器启动的基本环境。

☑　srm.conf：主要用于文件资源上的设定。

1.3.2　php.ini 文件的基本配置

php.ini 文件是 PHP 在启动时自动读取的配置文件。php.ini 是一个 ASCLL 文本文件，分为多个部分，每一部分包括相关的参数。每一部分的名称位于最前面的方括号内，接着是名称对数字，每一名称都独占一行。使用规则 PHP 代码，对参数名称非常敏感，不能包含有空格，但是参数可以是数字、字符串或者布尔逻辑数。分号位于每一行的开始，其作为指定标记，这就使选择使用或者不使用 PHP 的这些特性变得很方便，而无须通过删除该行来实现。对某特性进行注释（即添加分号），则该行将不会被编译执行。每次修改完 php.ini 文件，必须重新启动 Apache 服务器，以使新的设置生效。

你问我答：什么是 php.ini？

php.ini 是 PHP 的配置文件，用于加载各种函数库、设置错误级别和设置服务器的时间等。

在 Linux 操作系统中，php.ini 存储于/opt/lampp/etc/php.ini 文件夹下，而在 Windwos 操作系统中 php.ini 存储于系统盘的 WINDOWS 文件下。php.ini 文件的基本配置如表 1.1 所示。

表 1.1　php.ini 文件的基本配置

参　数	说　明	默 认 值
engine	设置解析器选项，关闭服务器表示嵌入的 PHP 代码不会被 Web 服务器解析。然而通常将服务器保持 On 状态	engine=On
short_open_tag	控制语法分析器是否识别快捷键<?...?>，类似于识别标准键<?php...?>	short_open_tag=On
register_globals	通常情况下可以将此变量设置为 Off，这样可以对通过表单进行的脚本攻击提供更为安全的防范措施	register_globals = On
include_path	设置 PHP 的搜索路径，这一参数可以接收系列的目录。当 PHP 遇到没有路径的文件提示时，它将会自动检测这些目录。需要注意的是，当某些选项允许多个值，应使用系统列表分隔符，在 Windows 下使用分号“;”，在 Linux 下使用冒号“:”	; UNIX: "/path1:/path2" ;include_path = ".:/php/includes" ; Windows: "\path1;\path2" ;include_path = ".;c:\php\includes"

续表

参　数	说　明	默 认 值
extension_dir	指定 PHP 的动态连接扩展库的目录	“\ext”目录下
extension	指定 PHP 启动时所加载的动态连接扩展库。PHP 的常用扩展库及其说明请参见表 1.2	PHP 的常用扩展库在初次安装配置后均被注释，需读者手动更改
file_uploads	设置是否允许通过 HTTP 上传文件	file_uploads=On
upload_tmp_dir	设置通过 HTTP 上传文件时的临时目录，如果为空，则使用系统的临时目录	upload_tmp_dir =空
upload_max_filesize	设置允许上传文件的大小，如 50MB，必须填写单位	upload_max_filesize=2MB
post_max_size	控制在采用 POST 方法进行一次表单提交中 PHP 所能够接收的最大容量。要上传更大的文件，则该值必须大于 upload_max_filesize 的值。如 upload_max_filesize=10MB，那么 upload_max_filesize 的值必须要大于 10MB	post_max_size = 8MB
max_input_time	以秒为单位对通过 POST、GET 以及 PUT 方式接收数据时间进行限制	max_input_time = 60
expose_php	设置用户的网页服务器容量所呈现的状态，设定 expose_php=Off，可以防止黑客把用户的网页服务器容量隐藏起来	expose_php = On
smtp	设置 SMTP 发信服务器的地址，仅支持 Windows 操作系统	SMTP = localhost
smtp_port	设置 SMTP 发信服务器的端口，仅支持 Windows 操作系统	smtp_port = 25
sendmail_from	设置正确格式的电子邮件地址，仅支持 Windows 操作系统	sendmail_from = me@example.com
safe_mode	设置是否打开安全模式	safe_mode = Off

表 1.2　PHP 常用扩展库及其说明

扩　展　库	说　明
php_ftp.dll	支持 FTP 函数库，可以实现客户机与服务器之间标准传送协议（FTP）
php_gd2.dll	支持图像处理函数库，支持.gif、.jpg、.png 等多种图像格式
php_imap.dll	支持 imap 电子邮件处理函数库
php_mssql.dll	支持 MsSQL 数据库
php_msql.dll	支持 mSQL 数据库
php_mysql.dll	支持 MySQL 数据库
php_oracle.dll	支持 Oracle 数据库
php_pdf.dll	支持 PDF 文件处理函数库
php_sockets.dll	支持 Sockets 处理函数库
php_zlib.dll	支持 zlib 文件压缩函数库

1.4　解决 PHP 的常见配置问题

程序运行出错是许多程序员最为头疼的问题，下面介绍 PHP 常见的配置问题。通过本节的学习，可以分清哪些错误是由于 PHP 环境配置不当产生的，从而避免不必要的时间资源浪费，高效地完成 Web 应用程

序的开发。

1.4.1　解决 Apache 服务器端口冲突

IIS 的默认端口号为 80，同 Apache 服务器默认端口号相同。由于采用了相同的端口号 80，因此，在运行网页时就会发生冲突。

如果用户机器上安装了 IIS，就需要修改 IIS 的默认端口，否则将导致 Apache 服务器无法正常工作。更改 IIS 的默认侦听端口 80，可以在 IIS 的管理器中进行设置，也可以停止 IIS 的服务。

用户也可以在安装 Apache 服务器时更改默认的端口号，从而解决两个服务器共用一个端口号而产生冲突的问题。

1.4.2　设置 PHP 的系统当前时间

由于 PHP 5.0 对 date()函数进行了重写，因此，目前的日期时间函数比系统时间少 8 个小时。在 PHP 语言中默认设置的是标准格林威治时间（即采用的是零时区），所以要获取本地当前的时间必须更改 PHP 语言中的时区设置。

其方法是：在 php.ini 文件中，找到[date]下的“;date.timezone =”选项，将该项修改为“date.timezone =Asia/Hong_Kong”，然后重新启动 Apache 服务器。

设置完成后，输出系统当前的时间就不会出现时差问题。

1.4.3　增加 PHP 扩展模块

增加 PHP 扩展模块也称为动态扩展，用来动态加载某个模块，它包含一个指令：extension。

在 Windows 操作系统下，加载模块的方法如下：

```
extension=php_java.dll
```

在 Linux 操作系统下，加载模块的方法如下：

```
extension=php_java.so
```

需要说明的是，只加载这一行代码并不一定能启用相关的扩展包，有时还需要确保在操作系统中安装相关的软件。例如，为启用 Java 支持，需要安装 JDK。

1.5　照猫画虎——基本功训练

1.5.1　基本功训练 1——测试 PHP 环境是否搭建成功

视频讲解：光盘\mr\01\lx\测试 PHP 环境是否搭建成功.exe

测试 AppServ 是否安装成功，打开 IE 浏览器，在地址栏中输入“http://localhost/”或者“http://127.0.0.1/”，如果打开如图 1.19 所示的页面，则说明 AppServ 安装成功。

技巧： 如果在安装时设置 Apache 的端口号是 82，那么在 IE 浏览器的地址栏中将输入“http://localhost:82/”或者“http://127.0.0.1:82/”来测试 AppServ 是否安装成功。切记一定要将 AppServ 安装在本地磁盘的根目录之下（如 E:\AppServ）。

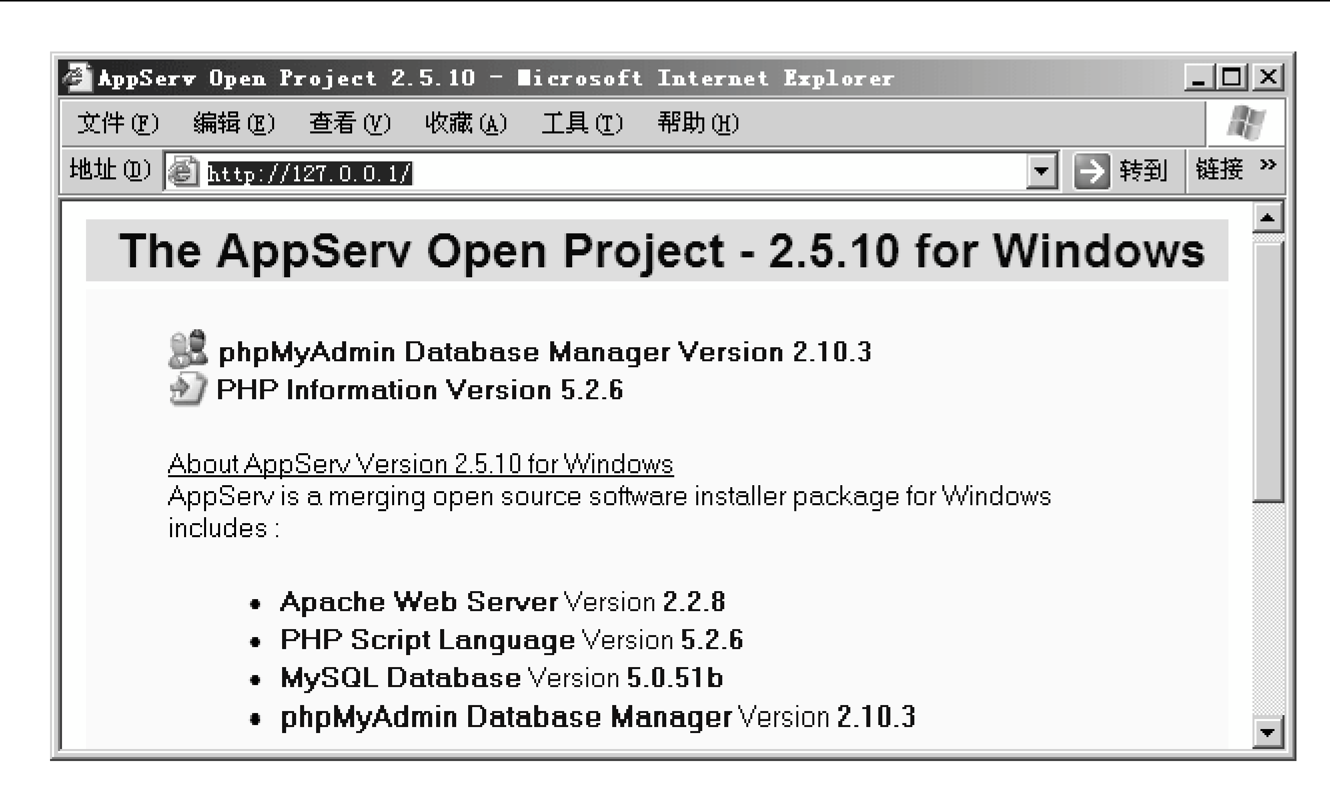


图 1.19　AppServ 测试页

照猫画虎：按照上面的操作过程，测试您安装的 AppServ 是否成功。（20 分）

1.5.2　基本功训练 2——第一次登录 phpMyAdmin 图形化管理工具

视频讲解：光盘\mr\01\lx\第一次登录 phpMyAdmin 图形化管理工具.exe

AppServ 安装完成并测试成功后，打开 IE 浏览器，在地址栏中输入“http://localhost/”或者“http://127.0.0.1/”，打开如图 1.19 所示的页面，单击页面最上方的 phpMyAdmin Database Manager Version 2.10.3。打开如图 1.20 所示的页面，它是图形化管理工具 phpMyAdmin 的登录窗口，输入在安装 AppServ 时设置的 MySQL 服务器的用户名和密码（用户名默认是 root，密码本机设置的是 111），即可登录到 phpMyAdmin 图形化管理工具中，如图 1.21 所示。



图 1.20　phpMyAdmin 的登录窗口　　图 1.21　phpMyAdmin 图形化管理工具界面

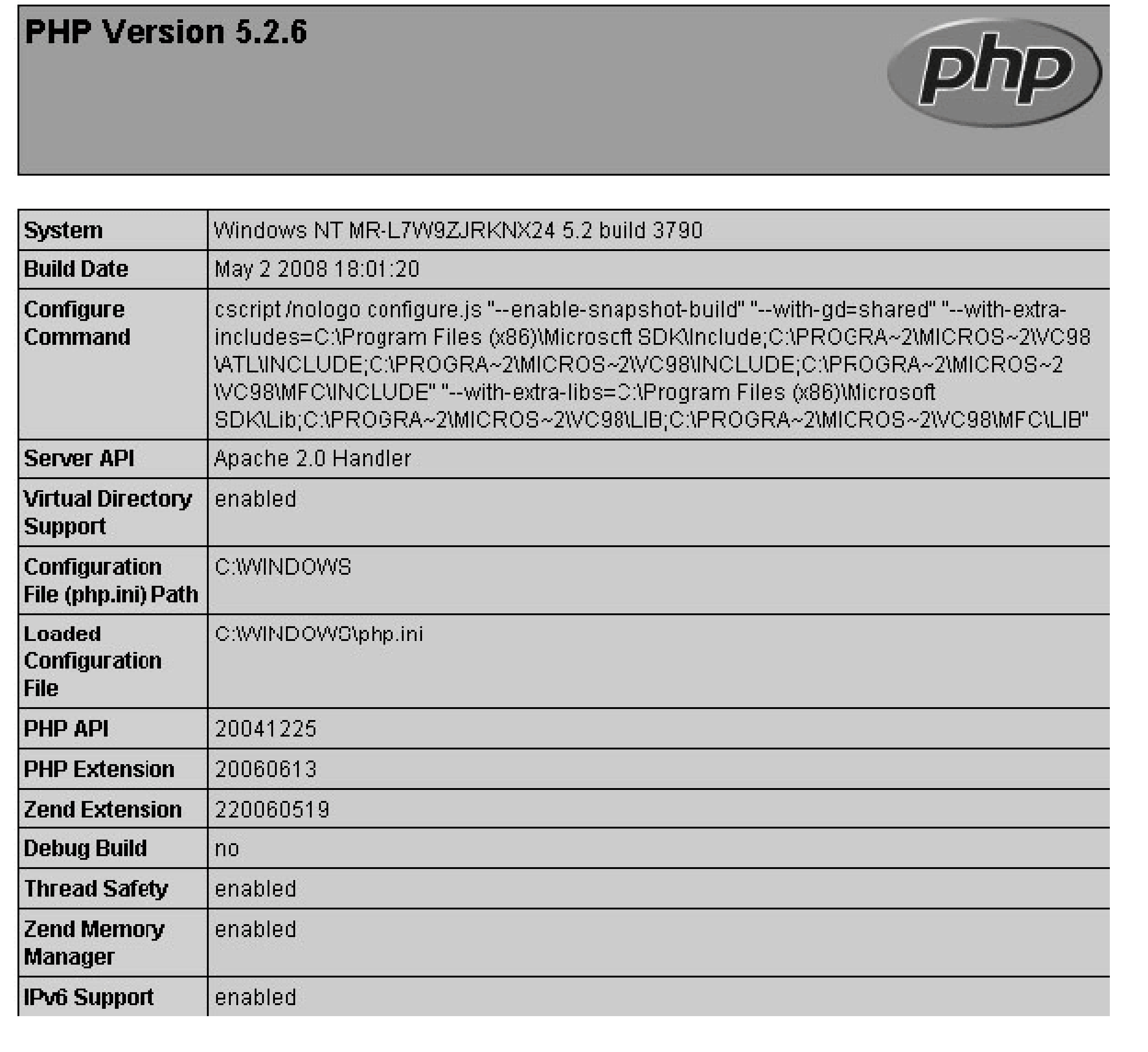
照猫画虎：打开本机上的 phpMyAdmin 图形化管理工具。（20 分）

1.5.3　基本功训练 3——浏览 PHP 环境配置说明

视频讲解：光盘\mr\01\lx\浏览 PHP 环境配置说明.exe

打开 IE 浏览器，在地址栏中输入“http://localhost/”或者“http://127.0.0.1/，打开如图 1.19 所示的页面，单击 PHP Information Version 5.2.6，打开如图 1.22 所示的页面，该页面就是 phpinfo()运行界面，在这个页面中读者可以查询到 PHP 环境的相关配置，以及 PHP 支持的扩展库。





PHP Version 5.2.6

System	Windows NT MR-L7W9ZJRKNX24 5.2 build 3790
Build Date	May 2 2008 18:01:20
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared" "--with-extra-includes=C:\Program Files (x86)\Microsoft SDK\Include;C:\PROGRA~2\MICROS~2\VC98\ATL\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\MFC\INCLUDE" "--with-extra-libs=C:\Program Files (x86)\Microsoft SDK\Lib;C:\PROGRA~2\MICROS~2\VC98\LIB;C:\PROGRA~2\MICROS~2\VC98\MFC\LIB"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled

图 1.22　phpinfo()运行界面

说明：由于该界面页码多，这里只是截取一部分。读者可以自己运行本机的 phpinfo 查看其他内容。

照猫画虎：打开本机上的 phpinfo()运行界面。（20 分）

1.5.4　基本功训练 4——解析 PHP 环境搭建的四大目录

视频讲解：光盘\mr\01\lx\解析 PHP 环境搭建的四大目录.exe

AppServ 安装完成后，整个目录默认安装在“E:\AppServ”，该目录下包含 4 个子目录，如图 1.23 所示。

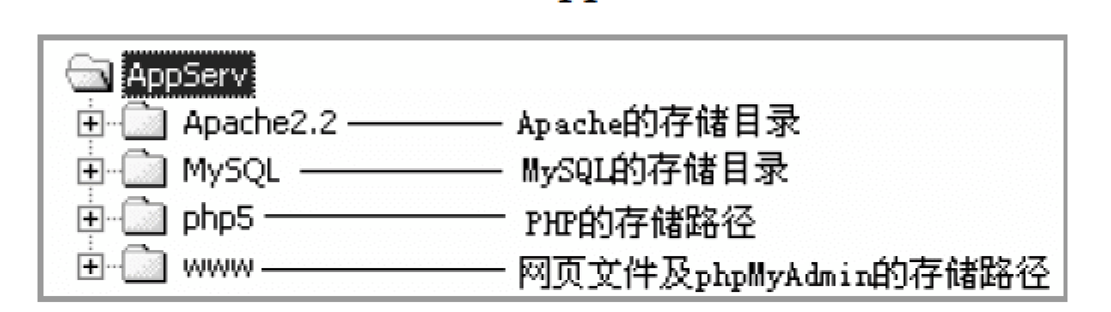


图 1.23　AppServ 目录结构

- ☑ 在 Apache2.2\conf\目录下有一个 httpd.conf 文件，它是 Apache 服务器的配置文件，在这个文件中可以修改 Apache 服务器的端口号、根目录等，所有有关 Apache 服务器的配置都在这个文件中完成。
- ☑ 在 MySQL 目录下有一个 my.ini 文件，它是 MySQL 服务器的配置文件，存储 MySQL 的配置信息。
- ☑ 在 MySQL\data 目录下存储的是数据库文件，所有程序使用的数据库都存储在这个文件夹下。
- ☑ 在 php5\ext 文件夹下存储的是 PHP 内置的函数类库，以.dll 的格式存储。而 PHP 的配置文件 php.ini 是存储在本机系统盘的 WINDOWS 文件夹下。
- ☑ www，程序运行的根目录，也就是说所有要运行的程序都必须存储在这个目录下。phpMyAdmin 图形化管理工具默认就存储在这个目录下。

照猫画虎：查看 AppServ 目录下的 4 个子目录，并简单了解目录下的文件。（20 分）

1.5.5 基本功训练 5——编辑第一个 PHP 程序

视频讲解：光盘\mr\01\lx\编辑第一个 PHP 程序.exe

实例位置：光盘\mr\01\zmhh\01

服务器环境配置完成后，接下来应用 Macromedia Dreamweaver 8 开发工具编写第一个 PHP 程序。编写第一个 PHP 程序的目的是熟悉 PHP 的书写规则和 Macromedia Dreamweaver 8 工具的基本使用方法。本实例很简单，就像大多数语言一样，输出一段欢迎信息。开发步骤如下：

（1）启动 Macromedia Dreamweaver 8 编辑器，选择“文件”/“新建”命令，打开“新建文档”对话框，在“常规”选项卡中选择“动态页”类别，在“动态页”列表框中选择 PHP 选项，如图 1.24 所示。

（2）单击“创建”按钮，即可成功创建一个动态的 PHP 页面。在新建页面的“代码”视图中编辑 PHP 代码，使用“设计”视图查看 HTML 效果。这里使用“代码”视图，并给该页面标题命名。如图 1.25 所示，标题显示的位置是在浏览器的左上角，稍后运行时就能看到效果。



图 1.24 “新建文档”对话框

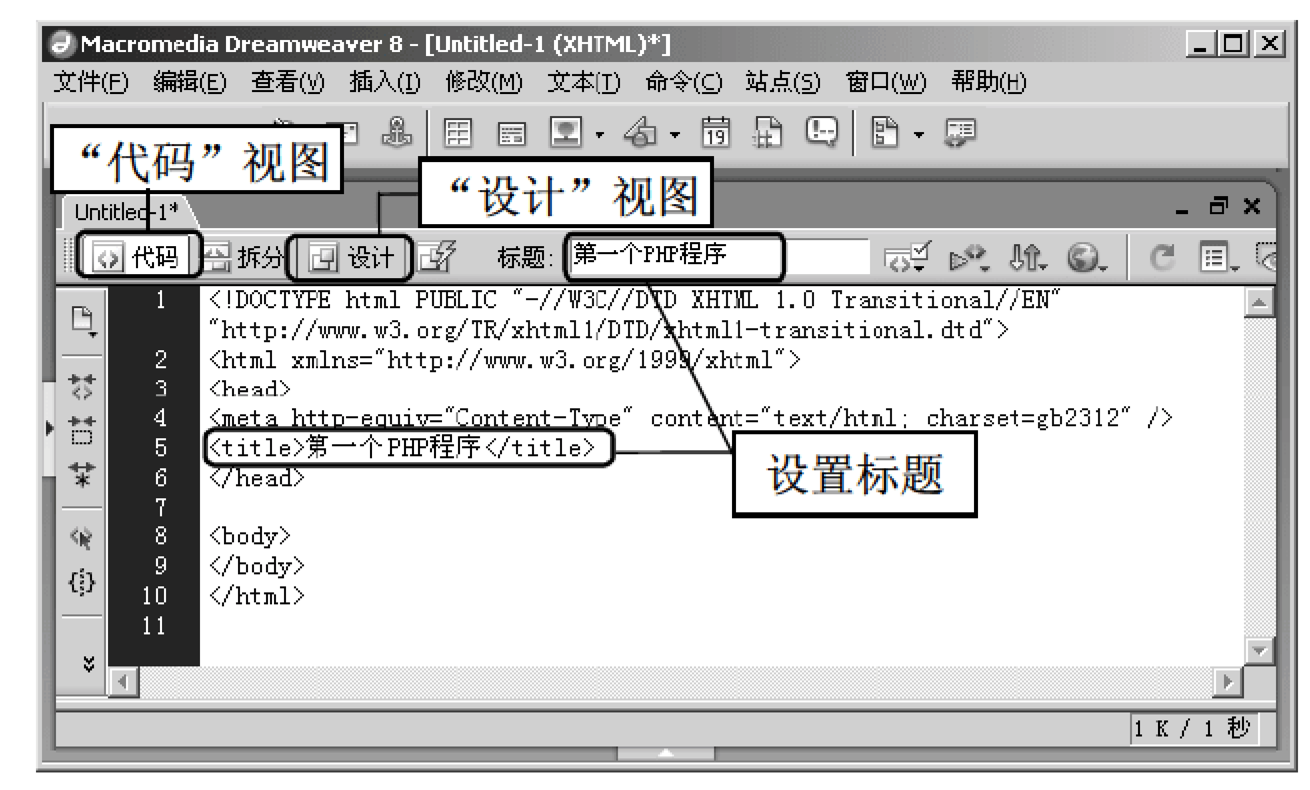


图 1.25 命名标题

（3）编写 PHP 代码。在<body>…</body>标签对中间开始编写 PHP 代码段，代码如下。

```
<?php
     echo  "欢迎加入 PHP 的大家庭！！";
?>
```

☑ “<?php”和“?>”是 PHP 的标记符。在这对标记符中的所有代码都被当作 PHP 代码来处理。除了这种表示方法外，PHP 还可以使用 ASP 风格的“<%”和 SGML 风格的“<?…?>”等，在以后的章节将会详细介绍。

☑ echo 语句是 PHP 中最常用的语句，主要用于将一个或多个字符串输出至网页，和 ASP 中的 response.write 以及 JSP 中的 out.print 是一个意思，就是将紧跟在后面的字符串或者变量值显示在页面中。每行代码都以分号“;”结尾。

注意： 在使用 echo 语句时，应该时刻牢记语句结束处的分号。在 PHP 中，分号是用来分隔语句的。丢失这个“;”是读者经常会犯的语法错误，同样也是读者在进行程序的错误处理时，首先应该注意的地方。

在 Macromedia Dreamweaver 8 中输入的 PHP 脚本程序如图 1.26 所示。

（4）将 PHP 动态页保存到服务器指定的目录以便解析。笔者指定的目录为 E:\AppServ\www\，将本页保存到 E:\AppServ\www\mr\01\zmhh\01\路径下，并将其命名为 index.php。



打开 IE 浏览器页面，在地址栏中输入“http://127.0.0.1/mr/01/zmhh/01/”，按 Enter 键后即可查看 index.php 页的执行结果，如图 1.27 所示。

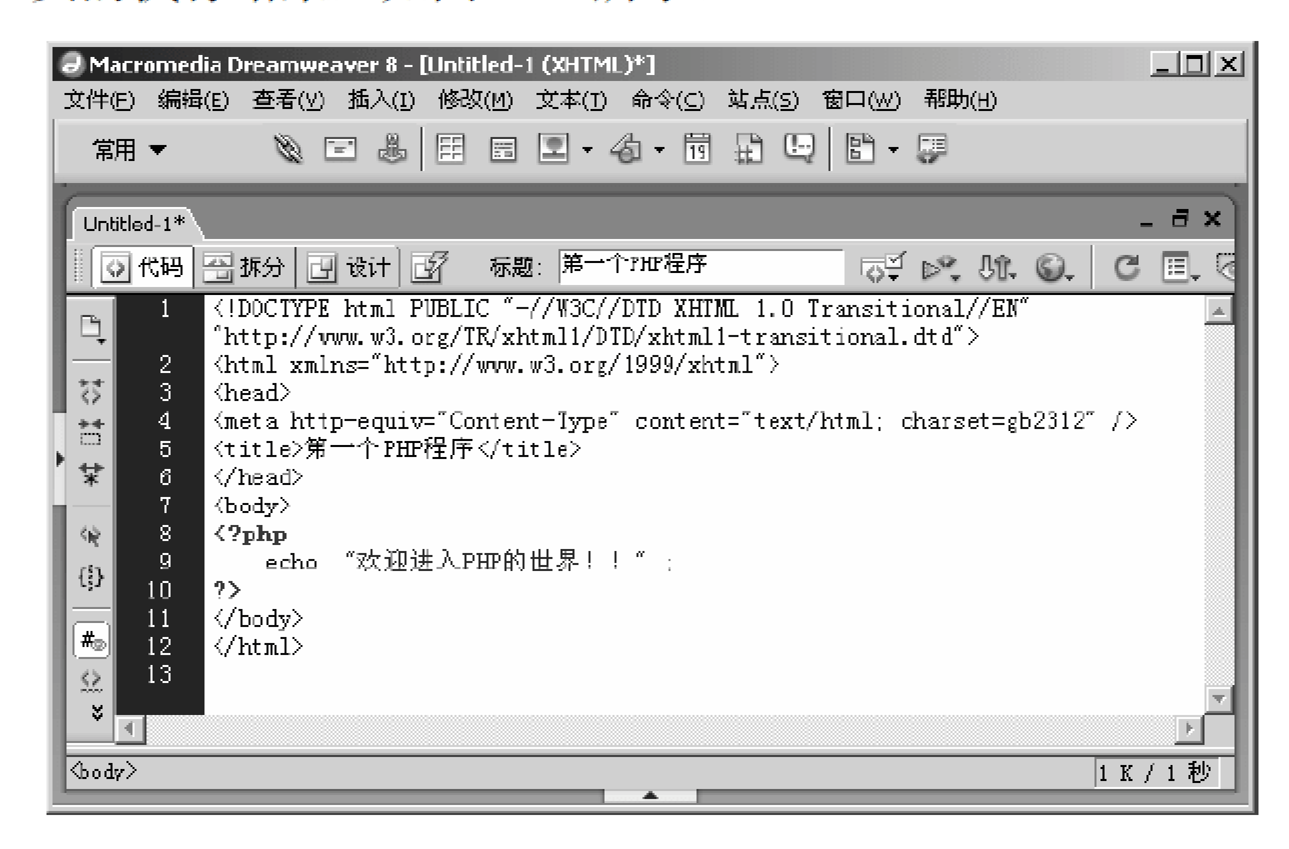


图 1.26　在开发工具中输入 PHP 脚本程序



图 1.27　PHP 页面运行结果

照猫画虎：尝试用 Macromedia Dreamweaver 8 编写一个 PHP 程序。（开发一个页面，使用 echo 语句输出字符串“恭喜您走上 PHP 的编程之路!”）（20 分）（实例位置：光盘\mr\01\zmhh\01_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

1.6　情景应用——拓展与实践

1.6.1　情景应用 1——输出系统的当前时间

视频讲解：光盘\mr\01\lx\输出系统的当前时间.exe

实例位置：光盘\mr\01\qjyy\01

在 1.4.2 小节中已经向大家介绍一种设置系统当前时间的方法，在这里将采用另外一种方式设置系统当前的时间，即直接在 date()函数之前通过 date_default_timezone_set()函数设置系统的时区，并输出系统当前的日期和时间。代码如下：

```
<?php
	echo "UTC 时间：".date("Y-m-d H:i:s");			//显示默认的 UTC 时间
	date_default_timezone_set("PRC");			//使用中华人民共和国的时区
	echo "<br>";
	echo "北京时间：".date("Y-m-d H:i:s");			//输出北京时间
	echo "<br>";
	echo "当前时区：".date_default_timezone_get();		//获取当前时区
?>
```

运行本实例，如果未对当前系统时区进行设置，则输出 UTC 时间；如果设置了 PHP 的时区，则输出北京时间，如图 1.28 所示。

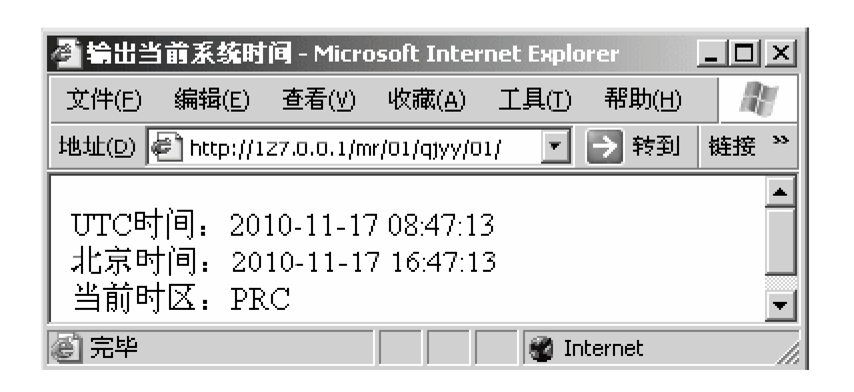


图 1.28　输出的当前系统时间

说明：date()是一个经常使用的日期、时间格式化处理函数，date 函数不仅可以用来表示日期，也可以用来表示时间。

DIY：输出系统当前时间，设置格式为 YYYY:MM:DD HH:MM:SS。（20 分）（实例位置：光盘\mr\01\qjyy\01_diy）

1.6.2　情景应用 2——输出“明日科技欢迎您！”

视频讲解：光盘\mr\01\lx\输出“明日科技欢迎您！”.exe

实例位置：光盘\mr\01\qjyy\02

在前面的学习中，已经向大家介绍了怎样应用 Macromedia Dreamweaver 8 开发工具来编写 PHP 程序。这里再通过一个简单的 PHP 程序让大家进一步熟悉 PHP 的书写规则和 Macromedia Dreamweaver 8 工具的基本使用方法。通过 PHP 脚本输出一段欢迎信息，并且以“Y 年 m 月 d 日 H 时 i 分 s 秒”的格式输出系统的当前时间。代码如下：

```
<?php
echo "明日科技欢迎您！";
echo date("Y 年 m 月 d 日  H 时 i 分 s 秒");
?>
```

运行结果如图 1.29 所示。

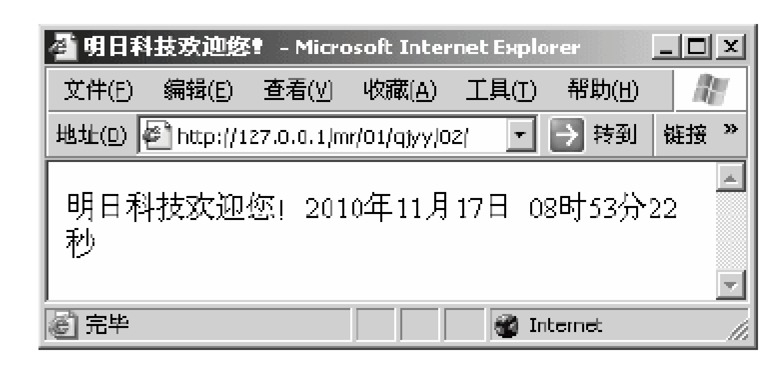


图 1.29　PHP 页面运行结果

DIY：应用 Macromedia Dreamweaver 8 开发工具来编写 PHP 程序，输出一句您喜欢的名言。（20 分）（实例位置：光盘\mr\01\qjyy\02_diy）

1.6.3　情景应用 3——输出一个漂亮的图片

视频讲解：光盘\mr\01\lx\输出一个漂亮的图片.exe

实例位置：光盘\mr\01\qjyy\03

这里将向大家介绍怎么通过 PHP 脚本输出一个漂亮的图片。具体实现过程如下：

（1）在实例根目录下新建一个 images 文件夹。

（2）将要输出的图片存储到新建的 images 文件夹下。

（3）打开 Macromedia Dreamweaver 8 编写 PHP 脚本，编写 echo 语句，输出一个完整的 img 图像标签，

进而完成图片的输出。代码如下：

```
<?php
	echo'<img src="images/03.jpg" width="98" height="98" />';
?>
```

（4）打开 IE 浏览器，在地址栏中输入“http://localhost/MR/01/qjyy/03/index.php”，按 Enter 键后，运行结果如图 1.30 所示。



图 1.30　图片在页面上的显示

DIY：尝试给页面上输出的图片加上提示信息（当鼠标置于图片上会显示提示信息）。提示：通过设置 img 标签中的 alt 属性来完成。（20 分）（实例位置：光盘\mr\01\qjyy\03_diy）

1.6.4　情景应用 4——更改 Apache 服务器的端口号为 82

视频讲解：光盘\mr\01\lx\更改 Apache 服务器的端口号为 82.exe

读者安装完 Apache 后再进行安装其他软件，有时会出现服务器端口号发生冲突，导致其他软件安装不成功，如安装迅雷。这时用户只需要将 Apache 服务器的端口号改成其他的即可。这里将 Apache 服务器端口号改为 82，具体做法如下：

（1）打开 Apache 目录下的 Apache2.2 子目录，找到 conf 文件夹（如 E:\AppServ\Apache2.2\conf）。

（2）通过记事本打开 httpd.conf 文件。

（3）按 Ctrl+F 键，搜索 80，定位到 Listen，将 80 端口修改为 82。

（4）选择“开始”/“管理工具”/“服务”命令，选择 Apache 2.2，单击鼠标右键，在弹出的快捷菜单中选择“重新启动”命令，重新启动 Apache 服务器。

（5）对端口号修改后的 Apache 进行测试，打开浏览器，在地址栏中输入“http://localhost:82/”，如果打开如图 1.31 所示的界面，则服务器端口修改成功，否则失败。



图 1.31　AppServ 测试页

说明：在 localhost 和 82 之间要用冒号（:）连接，初学者很容易忽略。

DIY：更改本机 Apache 服务器的端口号为 83。（20 分）

1.6.5　情景应用 5——Apache 端口号修改后运行第一个 PHP 程序

视频讲解：光盘\mr\01\lx\Apache 端口号修改后运行第一个 PHP 程序.exe

在完成 Apache 端口号修改后，重新运行第一个 PHP 程序。打开 IE 浏览器页面，在地址栏中输入

“http://localhost:82/mr/01/zmhh/01/”，按 Enter 键后即可查看 index.php 页的执行结果，如图 1.32 所示。

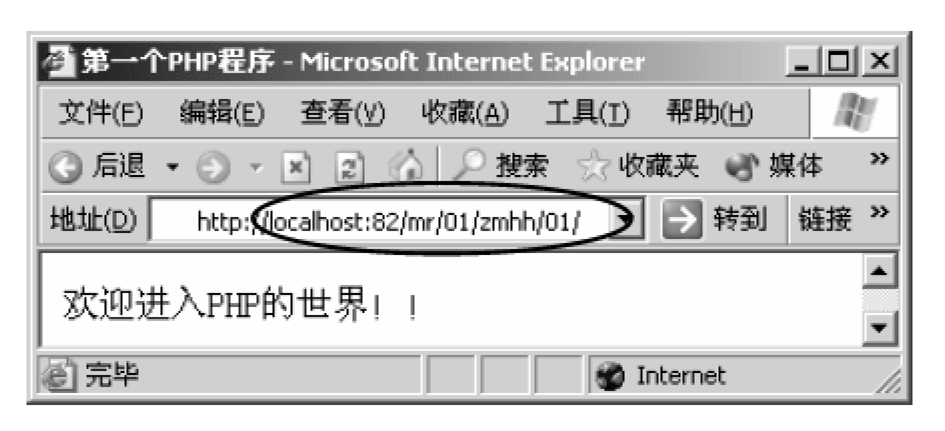


图 1.32　修改服务器端口号后，输出第一个 PHP 程序

DIY：根据 1.6.4 小节 DIY 更改 Apache 服务器的端口号，输出第一个 PHP 程序。（20 分）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

1.7　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．下面几个后台脚本编程语言中，（　　）个属于开源的软件。

　A．ASP　　B．PHP　　C．JSP　　D．CGI

2．在下面（　　）文件夹中能找到修改 Apache 服务器的端口号 httpd.conf 文件。

　A．conf　　B．bin　　C．error　　D．data

3．在本书中更改 PHP 语言中的时区设置有（　　）种方法。

　A．1　　B．2　　C．3

4．Apache 服务器默认的端口号为（　　）。

　A．80　　B．81　　C．82

5．用下面的代码，输出的内容是（　　）。

```
<?php
        echo "学习编程贵在坚持！";
?>
```

　A．输出内容为空　　B．学习编程贵在坚持！　　C．学习编程贵在坚持！！

二、填空题（每题 10 分，5 道题）

1．使用时间日期函数之前添加（　　　）不会出现时差。

2．php.ini 存储于系统盘的（　　　）文件夹中。

3．测试 AppServ 是否安装成功，在 IE 地址栏中输入（　　　）或者（　　　　　）。

4．AppServ 目录下有（　　）个子目录，分别是（　　）。

5．下面是输出时间的代码，请补充完整。

```
    <?php
        echo （          ）("Y-m-d H:i:s");
    ?>
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

1.8　行动指南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行动指南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）尝试开发一个页面，使用 echo 语句输出字符串“PHP 世界无限精彩”。	
		（2）尝试开发一个页面，使用 echo 语句输出一个 5*4 像素大小的表格。	
		（3）尝试开发一个页面，使用 echo 语句输出一张图片，在图片下方对图片进行解释说明。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

1.9 成功可以复制——PHP 开山鼻祖 Rasmus Lerdorf

成就：亲手设计了 PHP 语言，包括 PHP 1.0 和 PHP 2
国籍：加拿大
服务公司：University of Toronto、Bell Global Solutions
主要职务：产品经理、工程师 从事技术开发工作以及软件构架的研究
最高学历：博士
出生日期：1968 年

成功历程

PHP 的诞生和 Rasmus Lerdorf 的经历密不可分。Rasmus Lerdorf 于 1968 年生于格陵兰岛西部迪科斯岛上的 Godhavn/Qeqertarsuaq。他是 PHP 的开山鼻祖。在设计 PHP（作为现在的主流开发语言）之前，Rasmus Lerdorf 曾在 20 世纪 90 年代早期维护过新闻卫星通讯系统，并设计了用于验证数据正确性的错误验证协议。然后在 1991 年获得了两项加拿大的专利。在 1991—1993 年，他做了近 3 年的 UNIX 程序员，并成功地开发出了一个面向对象接口生成器。

Rasmus Lerdorf 为了要维护个人网页，用 C 语言开发了一些 CGI 工具程式集，来取代原先使用的 Perl 程式。最初这些工具程式用来显示 Rasmus Lerdorf 的个人履历，以及统计网页流量。他将这些程式和一些表单直译器整合起来，称为 PHP/FI。PHP/FI 可以和数据库连接，产生简单的动态网页程式。Rasmus Lerdorf 在 1995 年 6 月 8 日将 PHP/FI 公开释出，希望可以透过社群来加速程式开发与寻找错误。这个释出的版本命名为 PHP 2，已经有今日 PHP 的一些雏形，像是类似 Perl 的变量命名方式、表单处理功能以及嵌入到 HTML 中执行的能力。程式语法上也类似 Perl，有较多的限制，不过更简单、更有弹性。以后越来越多的网站使用了 PHP，并且强烈要求增加一些特性，如循环语句和数组变量等。

在新的成员加入开发行列之后，在 1995 年中，PHP 2.0 发布了。第二版定名为 PHP/FI（Form Interpreter）。PHP/FI 加入了对 MySQL 的支持，从此建立了 PHP 在动态网页开发上的地位。到了 1996 年底，有 15000 个网站使用 PHP/FI；时间到了 1997 年中，使用 PHP/FI 的网站数字超过 50000 个。而在 1997 年中，开始了第三版的开发计划，开发小组加入了 Zeev Suraski 及 Andi Gutmans，而第三版就定名为 PHP 3。2000 年，PHP 4.0 又问世了，其中增加了许多新的特性。

Rasmus Lerdorf 最为著名的成就是创造了最初的 PHP 引擎，并且成为了著名的开源开发者、演讲家和图书作者。

深度评价

也许在 PHP 研发时，Rasmus Lerdorf 也没有预想到会取得如此的关注与成功。但是 PHP 的成功并非偶然，它的问世凝聚了设计者的辛勤汗水与智慧。也许在 PHP 的设计道路上也是荆棘丛生，但是设计者仍然是以勇敢者的气魄，克服种种困难，最终迎接他们的是胜利的曙光。

第 2 堂课

PHP 基础

（视频讲解：140 分钟）

学习一门语言，首先要学习这门语言的语法，PHP 也不例外。本堂课开始学习 PHP 的基础知识，也是 PHP 的核心内容。不论是网站制作，还是应用程序开发，没有扎实的基本功是行不通的。开发一个功能模块，如果一边查函数手册一边写程序代码，要 15 天；但基础好的人只要 3～5 天，甚至更少的时间，孰优孰劣，立见分晓。为了将来应用 PHP 语言开发 Web 程序节省时间，现在就要认真地从基础学起，牢牢掌握 PHP 的基础知识。只有做到这一点，才能在以后的开发过程中事半功倍。

学习摘要：

- PHP 工作原理
- PHP 标记及注释
- PHP 常量
- PHP 变量
- PHP 数据类型
- PHP 运算符

2.1 PHP 工作原理

PHP 是 Hypertext Preprocessor（超文本预处理器）的缩写，是基于服务器端运行的脚本程序语言，实现数据库和网页之间的数据交互。

一个完整的 PHP 系统由以下几个部分构成。

- ☑ 操作系统：网站运行服务器所使用的操作系统。PHP 不要求操作系统的特定性，其跨平台的特性允许 PHP 运行在任何操作系统上，如 Windows、Linux 等。
- ☑ 服务器：搭建 PHP 运行环境时所选择的服务器。PHP 支持多种服务器软件，包括 Apache、IIS 等。
- ☑ PHP 包：实现对 PHP 文件的解析和编译。
- ☑ 数据库系统：实现系统中数据的存储。PHP 支持多种数据库系统，包括 MySQL、SQL Server、Oracle 及 DB2 等。
- ☑ 浏览器：浏览网页。由于 PHP 在发送到浏览器时已经被解析器编译成其他的代码，所以 PHP 对浏览器没有任何限制。

在图 2.1 中，完整地展示了用户通过浏览器访问 PHP 网站系统的全过程，从图中可以更加清晰地理清它们之间的关系。

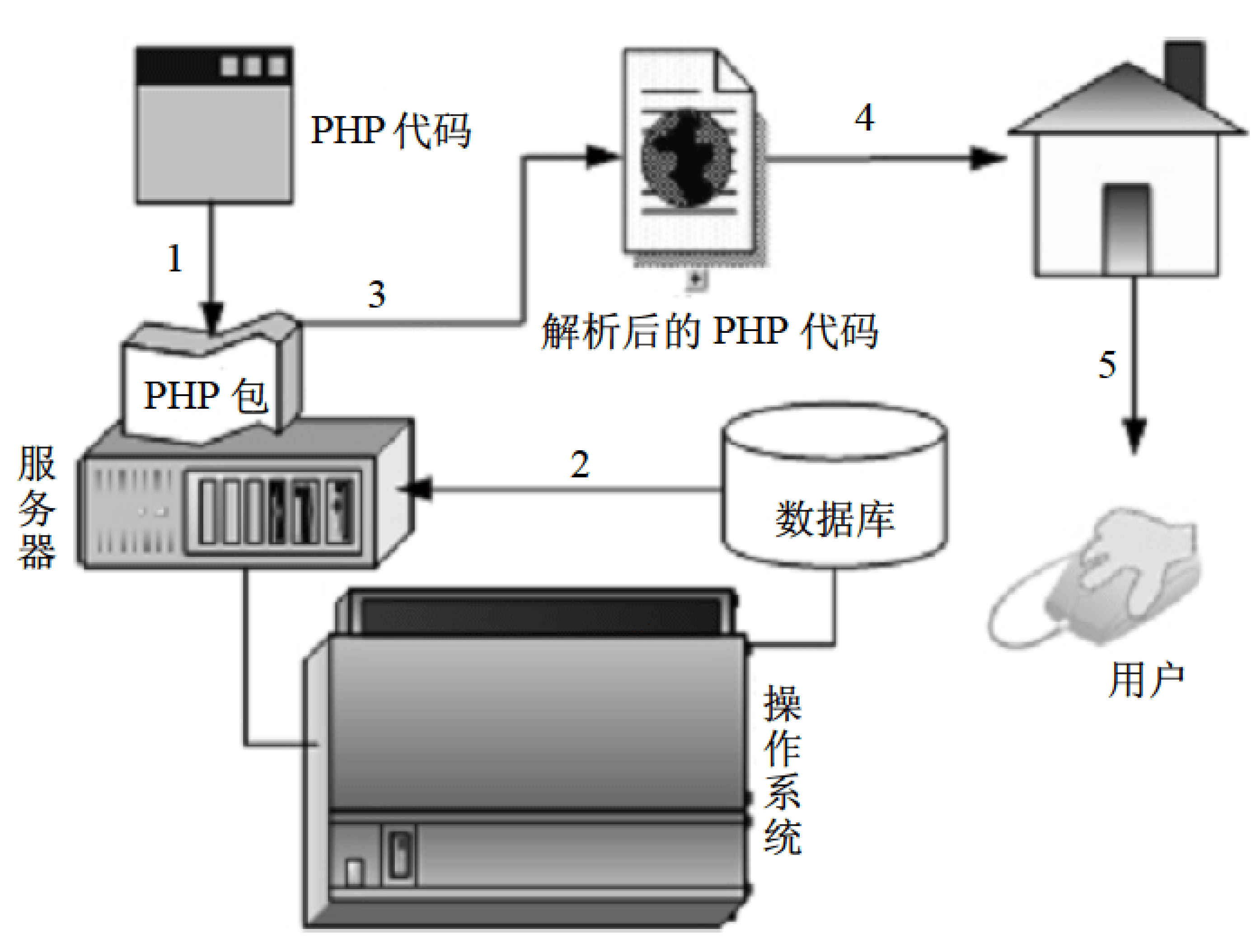


图 2.1　PHP 的工作原理

说明： 图 2.1 解析：

（1）PHP 的代码传递给 PHP 包，请求 PHP 包进行解析并编译。

（2）服务器根据 PHP 代码的请求读取数据库。

（3）服务器与 PHP 包共同根据数据库中的数据或其他运行变量，将 PHP 代码解析成普通的 HTML 代码。

（4）解析后的代码发送给浏览器，浏览器对代码进行分析获取可视化内容。

（5）用户通过访问浏览器浏览网站内容。



2.2　PHP 标记

所谓标记，就是为了便于与其他内容区分所使用的一种特殊标记，PHP 一共支持 4 种标记风格，下面分别进行介绍。

1．XML 标记风格

```
<?php
echo "这是 XML 标记风格";
?>
```

从上面的代码中可以看到，XML 风格标记是以“<?php”开始，以“?>”结尾的，中间包含的代码，就是 PHP 语言代码。推荐使用这种标记风格，因为它不能被服务器禁用，在 XML、XHTML 中都可以用。

2．脚本标记风格

```
<script language="php">
echo "这是脚本风格的标记";
</script>
```

脚本标记风格是以“<script …… ">”开头，以“</script>”结尾。

3．简短标记风格

```
<? echo "这是简短风格的标记"; ?>
```

如果想使用这种标记风格开发 PHP 程序，则必须保证 PHP 配置文件“php.ini”中的 short_open_tag 选项值设置为 On。

4．ASP 标记风格

```
<%
echo "这是 ASP 风格的标记";
%>
```

如果想使用这种标记风格开发 PHP 程序，则必须保证 PHP 配置文件“php.ini”中的 asp_tags 设置为 On。

2.3　代 码 注 释

注释可以理解为代码中的解释和说明，是程序中不可缺少的一个重要元素，使用注释不仅能够提高程序的可读性，而且还有利于程序的后期维护工作。注释不会影响程序的执行，因为在执行时，注释部分的内容不会被解释器执行。

2.3.1　使用 PHP 注释

PHP 的注释有 3 种风格，下面分别进行介绍。

1．C++风格的单行注释（//）

```
<?php
echo "使用 C++风格的注释";
```

```
//echo "这就是 C++风格的注释";
?>
```

运行结果为：使用 C++风格的注释

上面代码使用 echo 输出语句分别输出了“使用 C++风格的注释”和“这就是 C++风格的注释”，但是因为使用注释符号（//）将第二个输出语句注释掉了，所以不会被程序执行。

2．C 风格的多行注释（/*…*/）

```
<?php
    /*
    echo "这是第一行注释信息";
    echo "这是第二行注释信息";
    */
    echo "使用 C 风格的注释";
?>
```

运行结果为：使用 C 风格的注释

上面代码虽然使用 echo 输出语句分别输出了“这是第一行注释信息”、“这是第二行注释信息”和“使用 C 风格的注释”，但是因为使用了注释符号“/*…*/”将前面两个输出语句注释掉了，所以没有被程序执行。

3．Shell 风格的注释（#）

```
<?php
echo "这是 Shell 脚本风格的注释";     #这里的内容是看不到的
?>
```

运行结果为：这是 Shell 脚本风格的注释

因为使用了注释符号“#”，所以在#注释符号后面的内容是不会被程序执行的。

注意： 在使用单行注释时，注释内容中不要出现“?>”标志，因为解释器会认为这是 PHP 脚本，而去执行“?>”后面的代码。例如：

```
<?php
echo"这样会出错的！！！！！"     //不会看到?>会看到
?>
```

运行结果为：这样会出错的!!!!! 会看到?>

2.3.2　有效使用注释

程序注释是书写规范程序时很重要的一个环节。注释主要针对代码的解释和说明，用来解释脚本的用途、版权说明、版本号、生成日期、作者、内容等，有助于对程序的阅读理解。

合理使用注释有以下几项原则：

（1）注释语言必须准确、易懂、简洁。

说明： 错误的注释不但无益，反而有害。

（2）注释在编译代码时会被忽略，不会被编译到最后的可执行文件中，所以注释不会增加可执行文件的大小。

（3）注释可以书写在代码中的任意位置，但是一般写在代码的开头或者结束位置。



说明：避免在一行代码或表达式的中间插入注释，否则容易使代码可理解性变差。

（4）修改程序代码时，一定要同时修改相关的注释，以保持代码和注释的同步。

（5）在实际的代码规范中，要求注释占程序代码的比例达到 20%左右，即 100 行程序中包含 20 行左右的注释。

说明：防止没必要的重复注释信息。

（6）在程序块的结束行右方加注释标记，以表明某程序块的结束。

（7）避免在注释中使用缩写，特别是非常用缩写。

（8）注释与所描述内容进行同样的缩排，可使程序排版整齐，并方便注释的阅读与理解。

2.4　PHP 常量

常量可以理解为用于存储不经常改变的数据信息的量。常量的值被定义后，在程序的整个执行期间内，这个值都有效，并且不可再次对该常量进行赋值。

2.4.1　声明和使用常量

1．使用 define()函数声明常量

在 PHP 中使用 define()函数来定义常量，该函数的语法格式如下：

```
define(string constant_name,mixed value,case_sensitive=true)
```

define 函数的参数说明如表 2.1 所示。

表 2.1　define 函数的参数说明

参　　数	说　　明
constant_name	必选参数，常量名称，即标志符
value	必选参数，常量的值
case_sensitive	可选参数，指定是否大小写敏感，设定为 TRUE，表示不敏感

2．使用 constant()函数获取常量的值

获取指定常量的值和直接使用常量名输出的效果是一样的。但 constant()函数可以动态地输出不同的常量，在使用上要灵活、方便得多。

constant()函数的语法格式如下：

```
mixed constant(string const_name)
```

参数 const_name 为要获取常量的名称。如果成功则返回常量的值，失败则提示错误信息常量没有被定义。

3．使用 defined()函数判断常量是否已经被定义

defined()函数的语法格式如下：

```
bool defined(string constant_name);
```

参数 constant_name 为要获取常量的名称，成功则返回 TRUE，否则返回 FALSE。

例 2.01　下面举一个例子，使用 define()函数来定义名为 MESSAGE 的常量，使用 constant()函数来获

取该常量的值，最后再使用 defined()函数来判断常量是否已经被定义，代码如下。（**实例位置：光盘\mr\02\sl\2.01**）

```
<?php
/*使用 define()函数来定义名为 MESSAGE 的常量，并为其赋值为“能看到一次”，然后分别输出常量 MESSAGE 和 Message，因为没有设置 Case_sensitive 参数为 true，所以表示大小写敏感，因此执行程序时，解释器会认为没有定义该常量而输出提示，并将 Message 作为普通字符串输出 */
define("MESSAGE","能看到一次");
echo MESSAGE;
echo Message;
/*使用 define()函数来定义名为 COUNT 的常量，并为其赋值为“能看到多次”，并设置 Case_sensitive 参数为 true，表示大小写不敏感，分别输出常量 COUNT 和 Count，因为设置了大小写不敏感，因此程序会认为它和 COUNT 是同一个常量，同样会输出值*/
define("COUNT","能看到多次",true);
echo "<br>";
echo COUNT;
echo "<br>";
echo Count;
echo "<br>";
echo constant("Count");                    //使用 constant()函数来获取名为 Count 常量的值，并输出
echo "<br>";                               //输出空行符
echo (defined("MESSAGE"));     //判断 MESSAGE 常量是否已被赋值，如果已被赋值输出 1，如果未被赋值则返回 false
?>
```

运行结果如图 2.2 所示。



图 2.2　运行结果

2.4.2　预定义常量

PHP 中提供了很多预定义常量，可以获取 PHP 中的信息，但不能任意更改这些常量的值。预定义常量的名称及其功能如表 2.2 所示。

表 2.2　预定义常量的名称及其功能

常　量　名	功　　能
__FILE__	默认常量，PHP 程序文件名
__LINE__	默认常量，PHP 程序行数
PHP_VERSION	内建常量，PHP 程序的版本，如 3.0.8_dev
PHP_OS	内建常量，执行 PHP 解析器的操作系统名称，如 Windows
TRUE	该常量是一个真值（TRUE）

续表

常　量　名	功　　能
FALSE	该常量是一个假值（FALSE）
NULL	一个 null 值
E_ERROR	该常量指到最近的错误处
E_WARNING	该常量指到最近的警告处
E_PARSE	该常量指解析语法有潜在问题处
E_NOTICR	该常量为发生不寻常，但不一定是错误处

注意：__FILE__和__LINE__中的“__”是两条下划线，而不是一条“_”。表中以 E_开头的预定义常量，是 PHP 的错误调试部分。如需详细了解，请参考 error_reporting()函数。

例 2.02　下面使用预定义常量来输出 PHP 中的一些信息，代码如下。**（实例位置：光盘\mr\02\sl\2.02）**

```
<?php
echo "当前文件路径为：".__FILE__;                //使用__FILE__常量获取当前文件路径
echo "<br>";
echo "当前行数为：".__LINE__;                    //使用__LINE__常量获取当前所在行数
echo "<br>";
echo "当前 PHP 版本信息为：".PHP_VERSION;        //使用 PHP_VERSION 常量获取当前 PHP 版本
echo "<br>";
echo "当前操作系统为：".PHP_OS;                  //使用 PHP_OS 常量获取当前操作系统
?>
```

运行结果如图 2.3 所示。



图 2.3　使用预定义常量获取 PHP 信息

2.5　PHP 变量

前面介绍了常量，常量是不变的量，那么顾名思义，变量就是可以随时改变的量。变量主要用于存储临时数据信息，是编写程序中尤为重要的一部分。在定义变量时，通常要为其赋值，所以在定义变量的同时，系统会自动为该变量分配一个存储空间来存放变量的值。

2.5.1　声明变量

1．变量的定义

在 PHP 中变量的语法格式如下：

```
$变量名称=变量的值
```

PHP 中的变量命名应遵循以下规则：

☑ 在 PHP 中的变量名是区分大小写的。

☑ 变量名必须是以美元符号（$）开始。

☑ 变量名开头可以以下划线开始。

☑ 变量名不能以数字字符开头。

☑ 变量名可以包含一些扩展字符（如重音拉丁字母），但不能包含非法扩展字符（如汉字字符和汉字字母）。

正确的变量命名：

```
$name="mingri";          //定义一个变量，变量名为$name，变量值为 mingri
$_pwd="roof";            //定义一个变量，变量名为$pwd，变量值为 roof
$_123number=87665;       //定义一个变量，变量名为$123number，变量值为 87665
$_Class="roof";          //定义一个变量，变量名为$_Class，变量值为 roof
```

错误的变量命名：

```
$11112_var=11112;        //变量名不能以数字字符开头
$~%$_var="Lit";          //变量名不能包含非法字符
```

2．变量的赋值

变量的赋值有 3 种方式，下面分别进行介绍。

（1）直接赋值

直接赋值就是使用“=”直接将值赋给某变量，例如：

```
<?php
$name=mingri;
$number=30;
echo $name;
echo $number;
?>
```

运行结果为：

```
mingri
30
```

上例中分别定义了$name 变量和$number 变量，并分别为其赋了值，然后使用 echo 输出语句输出变量的值。

（2）传值赋值

传值赋值就是使用“=”将一个变量的值赋给另一个变量，例如：

```
<?php
$a=10;
$b=$a;
echo $a."<br>";
echo $b;
?>
```

运行结果为：

```
10
10
```

在上面的例子中，先定义了变量 a 并赋值为 10，然后又定义了变量 b，并设置变量 b 的值等于变量 a 的值，此时变量 b 的值也为 10。



（3）引用赋值

引用赋值就是一个变量引用另一个变量的值，例如：

```
<?php
$a=10;
$b=&$a;
$b=20;
echo $a."<br>";
echo $b;
?>
```

运行结果为：

```
20
20
```

仔细观察一下，“$b=&$a”中多了一个“&”符号，这就是引用赋值。当执行“$b=&$a”语句时，变量 b 将指向变量 a，并且和变量 a 共用同一个值。

当执行“$b=20”时，变量 b 的值发生了变化，此时由于变量 a 和变量 b 共用同一个值，所以当变量 b 的值发生变化时，变量 a 也随之发生变化。

2.5.2　变量作用域

变量的作用域就是指变量在哪些范围能被使用，在哪些范围不能被使用。PHP 中有 3 种变量作用域，分别为局部变量、全局变量和静态变量。

1．局部变量

局部变量就是在函数的内部定义的变量，其作用域是所在函数。

例 2.03　下面自定义一个名为 example()的函数，然后分别在该函数内部及函数外部定义并输出变量 a 的值，具体代码如下。**（实例位置：光盘\mr\02\sl\2.03）**

```
<?php
function example(){
	$a="hello php!";			//在自定义函数 example()中定义变量 a
	echo "在函数内部定义的变量 a 的值为：".$a."<br>";
}
example();
$a="hello china!";				//在函数外部定义变量 a
echo "在函数外部定义的变量 a 的值为：".$a."<br>";
?>
```

运行结果如图 2.4 所示。



图 2.4　局部变量示例运行结果

在本例中，只要了解局部变量的使用方法及意义即可，函数相关内容在第 3 堂课会详细讲解。

2. 全局变量

全局变量就是被定义在所有函数以外的变量，其作用域是整个 PHP 文件，但是在用户自定义函数内部是不可用的。想在用户自定义函数内部使用全局变量，要使用 global 关键词声明。

例 2.04 下面定义一个全局变量，并且在函数内部输出全局变量的值，具体代码如下。**（实例位置：光盘\mr\02\sl\2.04）**

```
<?php
$a="hello php!";                    //在自定义函数外部声明一个变量 a
function example(){                 //自定义一个函数，名为 example
    global $a;                      //使用 global 关键词声明并使用在函数外部定义的变量 a
    echo "在函数内部获得变量 a 的值为：".$a."<br>";
}
example();
?>
```

运行结果如图 2.5 所示。



图 2.5 全局变量示例运行结果

3. 静态变量

通过全局变量的理解，可以知道在函数内部定义的变量，在函数调用结束后，其变量将会失效。但有时仍然需要该函数内的变量有效，此时就需要将变量声明为静态变量，声明静态变量只需在变量前加“static”关键字即可。

例 2.05 下面分别在函数内声明静态变量和定义局部变量，具体代码如下。**（实例位置：光盘\mr\02\sl\2.05）**

```
<?php
function example(){
    static $a=10;           //定义静态变量
    $a+=1;
    echo "静态变量 a 的值为：".$a."<br>";
}
function zy(){
    $b=10;                  //定义局部变量
    $b+=1;
    echo "局部变量 b 的值为：".$b."<br>";
}
example();                  //一次执行该函数体
example();                  //二次执行该函数体
example();                  //三次执行该函数体
zy();                       //一次执行该函数体
zy();                       //二次执行该函数体
zy();                       //三次执行该函数体
?>
```

运行结果如图 2.6 所示。

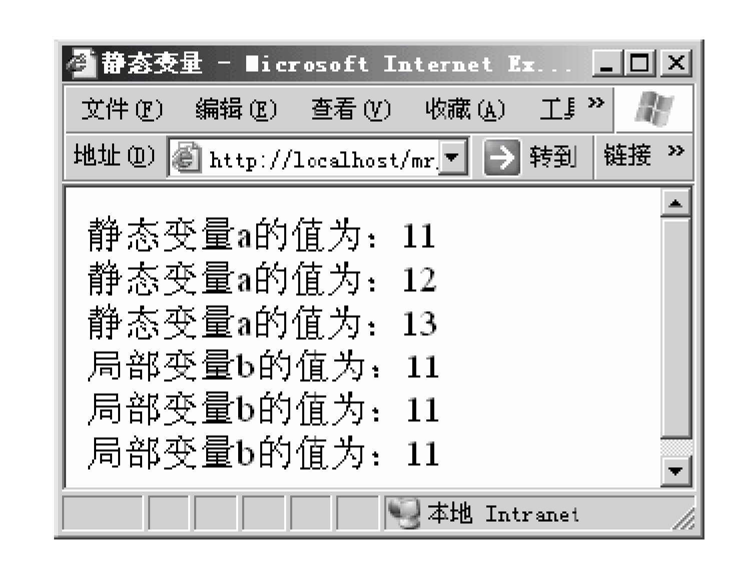


图 2.6　静态变量与普通变量的区别

2.5.3　可变变量

可变变量是一种独特的变量，这种变量的名称是由另外一个变量的值来确定的，声明可变变量的方法是在变量名称前加两个“$”符号。

声明可变变量的语法格式如下：

```
$$可变变量名称=可变变量的值
```

例 2.06　下面举例说明声明可变变量的方法，具体代码如下。**（实例位置：光盘\mr\02\sl\2.06）**

```
<?php
$a="mrkj";                  //定义变量
$$a="bccd";                 //声明可变变量，该变量名称为变量 a 的值
echo $a."<br>";             //输出变量 a
echo $$a."<br>";            //输出可变变量
echo $mrkj;                 //输出变量 mrkj
?>
```

运行结果如图 2.7 所示。



图 2.7　可变变量示例运行结果

2.6　PHP 数据类型

在计算机的世界里，计算机操作的对象是数据，而每一个数据都有其类型，具备相同类型的数据才可以彼此操作。PHP 的数据类型可以分为 3 种，即标量数据类型、复合数据类型和特殊数据类型。

2.6.1　标量数据类型

标量数据类型是数据结构中最基本的单元，只能存储一个数据。PHP 中标量数据类型包括 4 种，如表 2.3 所示。

表 2.3　标量数据类型

类　　型	说　　明
boolean（布尔型）	这是最简单的类型。只有两个值，真（TRUE）和假（FALSE）
string（字符串型）	字符串就是连续的字符序列，可以是计算机所能表示的一切字符的集合
integer（整型）	整型数据类型只能包含整数。这些数据类型可以是正数或负数
float（浮点型）	浮点数据类型用来存储数字，和整型不同的是它有小数位

下面对各个数据类型进行详细介绍。

1．布尔型（boolean）

布尔型是 PHP 中较为常用的数据类型之一。它保存一个真值（TRUE）或者假值（FALSE）。布尔型数据的用法如下：

```
<?php
$a=TRUE;
$c=FALSE;
?>
```

2．字符串型（string）

字符串是连续的字符序列，由数字、字母和符号组成。字符串中的每个字符只占用一个字节。字符包含以下几种类型：

☑　数字类型，如 1、2、3 等。

☑　字母类型，如 a、b、c、d 等。

☑　特殊字符，如#、$、%、^、&等。

☑　不可见字符，如\n（换行符）、\r（回车符）、\t（Tab 字符）等。

其中，不可见字符是比较特殊的一组字符，是用来控制字符串格式化输出的，在浏览器上不可见，只能看到字符串输出的结果。

例如，在下面的实例中通过不可见字符串输出一组字符串，程序代码如下：

```
<?php
echo  "PHP 从入门到精通\rASP 从入门到精通\nJSP 程序开发范例宝典\tPHP 函数参考大全"; //输出字符串
?>
```

运行本实例，在 IE 浏览器中不能直接查看到字符串的运行结果，只有通过“查看源文件”才能看到不可见字符串的运行结果。

在 PHP 中，定义字符串有 3 种方式：

☑　单引号（'）。

☑　双引号（"）。

☑　界定符（<<<）。

单引号和双引号是经常被使用的定义方式，定义格式如下：

```
$a ='字符串';
```

或

```
$a ="字符串";
```

两者的不同之处是：双引号中所包含的变量会自动被替换成实际数值，而在单引号中包含的变量则按普通字符串输出。



例 2.07　下面分别使用单引号、双引号、界定符输出变量的值，具体代码如下。（**实例位置：光盘\mr\02\sl\2.07**）

```
<?php
$a="你好！";
echo "$a"."<br>";                  //使用双引号输出变量
echo '$a'."<br>";                  //使用单引号输出$a
//使用界定符输出变量
echo <<<std
     $a
std;
?>
```

运行结果如图 2.8 所示。

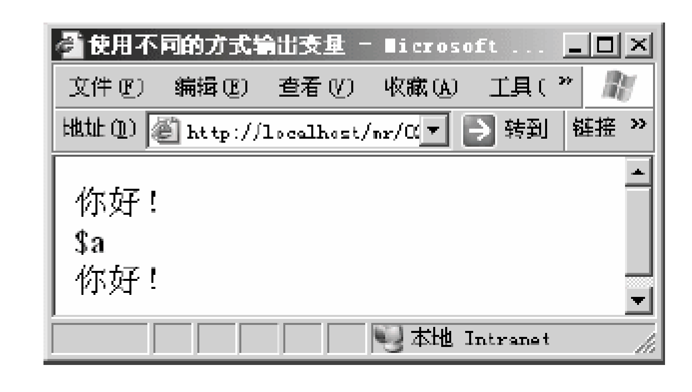


图 2.8　使用不同的方式输出变量的区别

> **注意：** 使用界定符输出字符串时，结束标识符必须单独另起一行，并且不允许有空格。如果在标识符前后有其他符号或字符，则会发生错误。

3．整型（integer）

整型数据类型只能包含整数。在 32 位的操作系统中，有效的范围是−2147483648～+2147483647。整型数可以用十进制、八进制和十六进制来表示。如果用八进制，数字前面必须加 0；如果用十六进制，则需要加 0x。

例 2.08　下面看一个实例，分别输出八进制、十进制和十六进制的结果，具体代码如下。（**实例位置：光盘\mr\02\sl\2.08**）

```
<?php
    $str1 = 1234;                  //八进制变量
    $str2 = 01234;                 //十进制变量
    $str3 = 0x1234;                //十六进制变量
    echo "数字 1234 不同进制的输出结果：<p>";
    echo "十进制的结果是：$str1<br>";
    echo "八进制的结果是：$str2<br>";
    echo "十六进制的结果是：$str3";
?>
```

运行结果如图 2.9 所示。

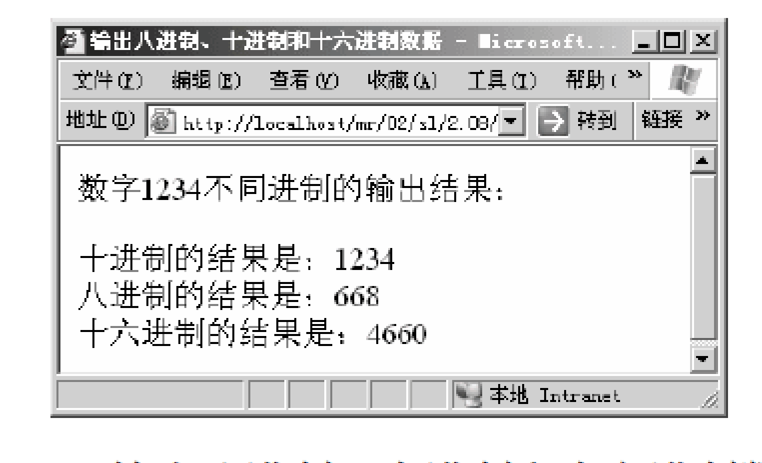


图 2.9　输出八进制、十进制和十六进制数据

注意：如果给定的数值超出了 int 类型所能表示的最大范围，将会被当作 float 型处理，这种情况叫做整数溢出。同样，如果表达式的最后运算结果超出了 int 的范围，也会返回 float 型。

4．浮点型（float）

浮点数据类型可以用来存储整数，也可以保存小数。它提供的精度比整数大得多。在 32 位的操作系统中，有效的范围是 1.7E−308～1.7E+308。在 PHP 4.0 以前的版本中，浮点型的标识为 double，也叫双精度浮点数，两者没什么区别。

浮点型数据默认有两种书写格式，一种是标准格式，例如：

```
3.1415
0.333
−35.8
```

还有一种是科学计数法格式，例如：

```
3.58E1
849.72E−3
```

例如：

```
<?php
$a=1.036;
$b=2.035;
$c=3.58E1;          //该变量的值为 3.58*1²
?>
```

注意：浮点型的数值只是一个近似值，所以要尽量避免浮点型之间比较大小，因为最后的结果往往是不准确的。

2.6.2　复合数据类型

复合数据类型包括两种，即 array（数组）和 object（对象）。

1．数组（array）

数组是一组数据的集合，它把一系列数据组织起来，形成一个可操作的整体。数组中可以包括很多数据：标量数据、数组、对象、资源，以及 PHP 中支持的其他语法结构等。

数组中的每个数据称为一个元素，每个元素都有一个唯一的编号，称为索引。元素的索引只能由数字或字符串组成。元素的值可以是多种数据类型。定义数组的语法格式如下：

```
$array[key] = "value";
```

或

```
$array(key1 => value1, key2 => value2……)
```

其中参数 key 是数组元素的索引，value 是数组元素的值。

例 2.09　下面举一个简单的数组应用示例，具体代码如下。**（实例位置：光盘\mr\02\sl\2.09）**

```
<?php
$array[0]="明日科技";                                    //定义$array 数组的第一个元素
$array[1]="编程词典";                                    //定义$array 数组的第二个元素
$array[2]="编程无忧";                                    //定义$array 数组的第 3 个元素
$number=array(0=>'明日科技',1=>'编程词典',2=>'编程无忧');    //定义$number 数组的所有元素
echo $array[0]."<br>";                                   //输出$array 数组的第一个元素值
```

```
echo $number[1];                                //输出$number 数组的第二个元素值
?>
```

运行结果如图 2.10 所示。



图 2.10　数组应用

注意：本书第 5 堂课对数组进行了详细讲解，此处读者只需了解什么是数组即可。

2．对象（object）

现在的编程语言用到的方法有两种，即面向过程和面向对象。在 PHP 中，用户可以自由使用这两种方法。有关面向对象的技术可以参考本书后面的内容。

2.6.3　特殊数据类型

特殊数据类型包括两种，即 resource（资源）和 null（空值）。

1．资源（resource）

资源是由专门的函数来建立和使用的。它是一种特殊的数据类型，并由程序员分配。在使用资源时，要及时地释放不需要的资源。如果程序员忘记了释放资源，系统自动启用垃圾回收机制，避免内存消耗殆尽。

2．空值（null）

空值，顾名思义，表示没有为该变量设置任何值，另外，空值（null）不区分大小写，null 和 NULL 效果是一样的。被赋予空值的情况有以下 3 种：

- ☑　没有赋任何值。
- ☑　被赋值为 null。
- ☑　被 unset()函数处理过的变量。

下面分别对这 3 种情况举例说明，具体代码如下：

```
<?php
$a;                 //没有赋值的变量
$b=NULL;            //被赋空值的变量
$c=3;
unset($c);          //使用 unset()函数处理后，$c 的值为空
?>
```

2.6.4　转换数据类型

PHP 中的类型转换和 C 语言一样，非常简单。在变量前面加上一个小括号，并把目标数据类型写在小括号中即可。

PHP 中允许转换的类型如表 2.4 所示。

表 2.4　PHP 中允许转换的类型

转 换 函 数	转 换 类 型	举　　例
(boolean),(bool)	将其他数据类型强制转换成布尔型	$a=1; $b=(boolean)$a; $b=(bool)$a;
(string)	将其他数据类型强制转换成字符串型	$a=1; $b=(string)$a;
(integer),(int)	将其他数据类型强制转换成整型	$a=1; $b=(int)$a; $b=(integer)$a;
(float),(double),(real)	将其他数据类型强制转换成浮点型	$a=1; $b=(float)$a; $b=(double)$a; $b=(real)$a;
(array)	将其他数据类型强制转换成数组	$a=1; $b=(array)$a;
(object)	将其他数据类型强制转换成对象	$a=1; $b=(object)$a;

在进行类型转换的过程中应该注意以下几点：

☑　转换成 boolean 型

null、0 和未赋值的变量或数组，会被转换为 FALSE，其他的为真。

☑　转换成整型

布尔型的 FALSE 转为 0，TRUE 转为 1。

浮点型的小数部分被舍去。

字符串型。如果以数字开头，就截取到非数字位，否则输出 0。

当字符串转换为整型或浮点型时，如果字符是以数字开头的，就会先把数字部分转换为整型，再舍去后面的字串。如果数字中含有小数点，则会取到小数点前一位。

2.6.5　检测数据类型

PHP 中提供了很多检测数据类型的函数，可以对不同类型的数据进行检测，判断其是否属于某个类型。检测数据类型的函数如表 2.5 所示。

表 2.5　检测数据类型的函数

函　　数	检 测 类 型	举　　例
is_bool	检查变量是否为布尔类型	is_book($a);
is_string	检查变量是否为字符串类型	is_string($a);
is_float/is_double	检查变量是否为浮点类型	is_float($a); is_double($a);
is_integer/is_int	检查变量是否为整数	is_integer($a); is_int($a);
is_null	检查变量是否为 null	is_null($a);
is_array	检查变量是否为数组类型	is_array($a);
is_object	检查变量是否为一个对象类型	is_object($a);
is_numeric	检查变量是否为数字或由数字组成的字符串	is_numeric($a);

例 2.10　下面通过几个检测数据类型的函数来检测相应的字符串类型，具体代码如下。（**实例位置：光盘\mr\02\sl\2.10**）

```
<?php
$a=true;
$b="你好 PHP";
$c=123;
echo "变量是否为布尔型：".is_bool($a)."<br>";                    //检测变量是否为布尔型
```



```
echo "变量是否为字符串型：".is_string($b)."<br>";            //检测变量是否为字符串型
echo "变量是否为整型：".is_int($c)."<br>";                  //检测变量是否为整型
echo "变量是否为浮点型：".is_float($c)."<br>";               //检测变量是否为浮点型
?>
```

运行结果如图 2.11 所示。



图 2.11　检测变量数据类型

由于变量 C 不是浮点型，所以返回值为空值。

2.7　PHP 的运算符

PHP 运算符包括算术运算符、字符串运算符、赋值运算符、逻辑运算符等，下面分别对各运算符进行介绍。

2.7.1　算术运算符

算术运算符主要用于处理算术运算操作，常用的算术运算符如表 2.6 所示。

表 2.6　常用的算术运算符

名　　称	操　作　符	实　　例
加法运算	+	$a + $b
减法运算	-	$a-$b
乘法运算	*	$a * $b
除法运算	/	$a / $b
取余数运算	%	$a % $b

说明：在算术运算符中使用“%”求余，如果被除数（$a）是负数，那么取得的结果也是一个负值。

例 2.11　下面举例说明上述几种运算符的使用方法，具体代码如下。（**实例位置：光盘\mr\02\sl\2.11**）

```
<?php
$a=5;                              //定义变量 a 的值为 5
$b=9;                              //定义变量 b 的值为 9
echo "a=".$a."<br>";               //输出变量 a 的值
echo "b=".$b."<br>";               //输出变量 b 的值
echo "a+b=".$a+$b."<br>";          //计算变量 a 加变量 b 的值
echo "b-a=".$b-$a."<br>";          //计算变量 b 减变量 a 的值
echo "a*b=".$a*$b."<br>";          //计算变量 a 乘以变量 b 的值
echo "a/b=".$a/$b."<br>";          //计算变量 a 除以变量 b 的值
```

```
echo "b%a=".$b%$a."<br>";                  //计算变量 b 和变量 a 余数
?>
```

运行结果如图 2.12 所示。



图 2.12　算术运算符示例运行结果

2.7.2　字符串运算符

字符串运算符主要用于处理字符串的相关操作，在 PHP 中字符串运算符只有一个，那就是“.”，该运算符用于将两个字符串连接起来，结合到一起形成一个新的字符串。应用格式如下：

```
$a.$b
```

此运算符在前面的例子中已经使用过，如例 2.11 中的：

```
echo "a=".$a."<br>";
```

此处使用了两个字符串运算符“.”，第一处是将字符串文字“a=”与变量$a 的值连接，以输出“a=5”的字符串；第二处是将刚刚获得的“a=5”的字符串与换行符“
”连接，如果不加此符号，那么所有结果将会在一行输出。

2.7.3　赋值运算符

赋值运算符主要用于处理表达式的赋值操作，PHP 中提供了很多赋值运算符，其用法及意义如表 2.7 所示。

表 2.7　常用赋值运算符的用法及意义

操　作	符　号	实　例	展开形式	意　义
赋值	=	$a=b	$a=3	将右边的值赋给左边
加	+=	$a+= b	$a=$a + b	将右边的值加到左边
减	-=	$a-= b	$a=$a-b	将右边的值减到左边
乘	*=	$a*= b	$a=$a * b	将左边的值乘以右边
除	/=	$a/= b	$a=$a / b	将左边的值除以右边
连接字符	.=	$a.= b	$a=$a. b	将右边的字符加到左边
取余数	%	$a%= b	$a=$a % b	将左边的值对右边取余数

下面举一个非常简单的赋值运算符的例子，就是为变量赋值：

```
$a=5;
```

此处应用“=”运算符，为变量 a 赋值，下面再举一个复杂一点的示例，代码如下：

```
<?php
$a=5;              //使用“=”运算符为变量 a 赋值
```



```
$b=10;              //使用“=”运算符为变量 b 赋值
$a*=$b;             //使用“*=”运算符获得变量 a 乘以变量 b 的值，并赋给变量 a
echo $a;            //输出重新赋值后变量 a 的值
?>
```

运行结果为：50

2.7.4　位运算符

位运算符是指对二进制位从低位到高位对齐后进行运算。PHP 中的位运算符如表 2.8 所示。

表 2.8　位运算符

符　　号	作　　用	举　　例
&	按位与	$m & $n
|	按位或	$m | $n
^	按位异或	$m ^ $n
~	按位取反	$m ~ $n
<<	向左移位	$m << $n
>>	向右移位	$m >> $n

例 2.12　下面使用位运算符对变量中的值进行位运算操作，实例代码如下。（**实例位置：光盘\mr\02\sl\2.12**）

```
<?php
$m = 8 ;                    //运算时会将 8 转换为二进制码 1000
$n = 12 ;                   //运算时会将 12 转换为二进制码 1100
$mn = $m&$n ;               //将 1000 和 1100 做与操作后转换为十进制码
echo $mn ."<br>";           //输出转换结果
$mn = $m | $n ;             //将 1000 和 1100 做或操作后转换为十进制码
echo $mn ."<br>";           //输出转换结果
$mn = $m ^ $n ;             //将 1000 和 1100 做异或操作后转换为十进制码
echo $mn ."<br>";           //输出转换结果
$mn = ~$m ;                 //将 1000 做非操作后转换为十进制码
echo $mn ."<br>";           //输出转换结果
?>
```

运行结果为：8　　12　　4　　−9

2.7.5　递增或递减运算符

递增运算符“++”和递减运算符“−−”与算术运算符有些相同，都是对数值型数据进行操作。但算术运算符适合在两个或者两个以上不同操作数的场合使用，当只有一个操作数时，就可以使用“++”或者“−−”运算符。

例 2.13　下面举个简单的例子，来加深对递增和递减运算符的理解，具体代码如下。（**实例位置：光盘\mr\02\sl\2.13**）

```
<?php
$a=10;
$b=5;
$c=8;
```

```
$d=12;
echo  "a=".$a."  b=".$b."  c=".$c."  d=".$d."<br>";   //输出上面 4 个变量的值， 是空格符
echo "++a=".++$a."<br>";              //计算变量 a 自加的值
echo "b++=".$b++."<br>";              //计算变量 b 自加的值
echo "--c=".--$c."<br>";              //计算变量 c 自减的值
echo "d--=".$d--."<br>";              //计算变量 d 自减的值
?>
```

运行结果如图 2.13 所示。

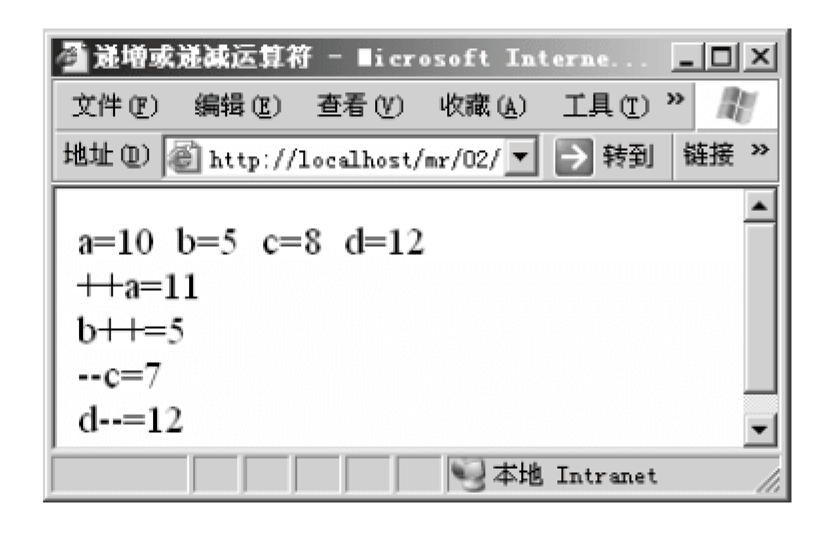


图 2.13 递增和递减运算符示例运行结果

你问我答：例 2.13 中变量$b 自加和$d 自减后的值为什么没变？

当运算符位于变量前时（++$a），先自加，然后再返回变量的值；当运算符位于变量后时（$a++），先返回变量的值，然后再自加，虽然输出的值没有变化，但实际上“b++”的值已经为 6，而“d--”的值已变为 11。

2.7.6 逻辑运算符

逻辑运算符用于处理逻辑运算操作，是程序设计中一组非常重要的运算符。PHP 的逻辑运算符如表 2.9 所示。

表 2.9 PHP 的逻辑运算符

运　算　符	实　　例	结 果 为 真
&&或 and（逻辑与）	$m and $n 或$m && $n	当$m 和$n 都为真或假时，返回 TRUE 或 FALSE 当$m 和$N 有一个为假时，返回 FLASE
||或 or（逻辑或）	$m || $n 或$m or $n	当$m 和$n 都为真或假时，返回 TRUE 或 FALSE 当$m 和$N 有一个为真时，返回 TRUE
xor（逻辑异或）	$m xor $n	当$m 和$n 都为真或假时，返回 TRUE 或 FALSE 当$m 和$N 有一个为真时，返回 TRUE
!（逻辑非）	!$m	当$m 为假时返回 TRUE，当$m 为真时返回 FALSE

例 2.14 下面分别使用几种逻辑运算符举一个小示例，具体代码如下。**（实例位置：光盘\mr\02\sl\2.14）**

```
<?php
$a=true;
$b=true;
if($a and $b){          //使用 and 判断变量 a 和变量 b
      echo "变量 a 和变量 b 的值都为真";
```

```
}else{
	echo "变量 a 和变量 b 的值都为假";
}
?>
```

运行结果如图 2.14 所示。



图 2.14　使用 and 判断变量的真假

本例中使用 if…else 流程语句判断其返回值，如果为真则输出 TRUE，如果为假则输出 FALSE。因为变量 a 和变量 b 的值都为 TRUE，所以执行 if 语句的第一个语句块，返回值为 TRUE。

说明：关于 if…else 流程语句在第 4 堂课详细讲解，此处只要了解逻辑运算符的运算规律即可。

2.7.7　比较运算符

比较运算符主要用于比较两个数据的值，返回值为一个布尔类型。PHP 中的比较运算符如表 2.10 所示。

表 2.10　PHP 中的比较运算符

运　算　符	实　　例	结 果
<	小于	$m<$n，当$m 小于$n 时，返回 TRUE，否则返回 FALSE
>	大于	$m>$n，当$m 大于$n 时，返回 TRUE，否则返回 FALSE
<=	小于等于	$m<=$n，当$m 小于等于$n 时，返回 TRUE，否则返回 FALSE
>=	大于等于	$m>=$n，当$m 大于等于$n 时，返回 TRUE，否则返回 FALSE
= =	相等	$m= =$n，当$m 等于$n 时，返回 TRUE，否则返回 FALSE
!=	不等	$m!=$n，当$m 不等于$n 时，返回 TRUE，否则返回 FALSE
= = =	恒等	$m= = = $n，当$m 等于$n，并且数据类型相同时，返回 TRUE，否则返回 FALSE
!==	非恒等	$m!= =$n，当$m 不等于$n，并且数据类型不相同时，返回 TRUE，否则返回 FALSE

注：这里面= = =和!= =不太常见。

例 2.15　下面使用比较运算符来比较两个变量的值，具体代码如下。（**实例位置：光盘\mr\02\sl\2.15**）

```
<?php
$a=100;
$b=200;
echo "a=".$a."  b=".$b."<br>";
echo "a < b 的返回值为： ";
echo var_dump($a<$b)."<br>";			//比较 a 是否小于 b
echo "a >= b 的返回值为： ";
echo var_dump($a>=$b)."<br>";			//比较 a 是否大于等于 b
echo "a == b 的返回值为： ";
echo var_dump($a==$b)."<br>";			//比较 a 是否等于 b
echo "a != b 的返回值为： ";
```

```
echo var_dump($a!=$b)."<br>";            //比较 a 是否不等于 b
?>
```

运行结果如图 2.15 所示。

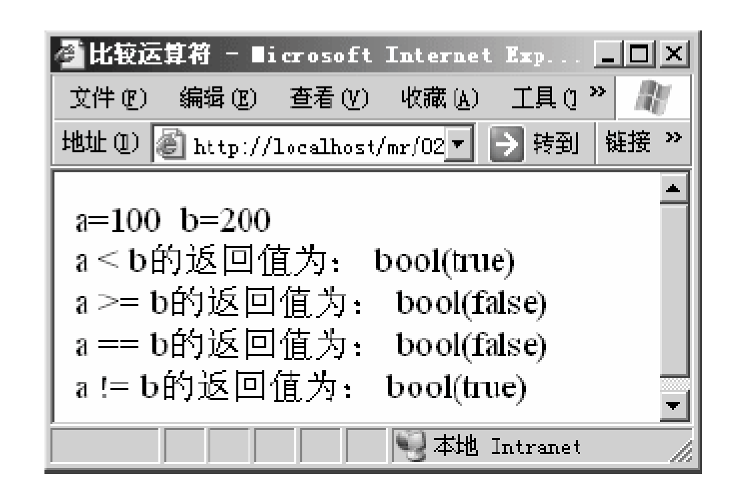


图 2.15　比较运算符示例运行结果

本例中使用 var_dump 函数来输出变量对比后返回的值，此处只需了解比较运算符比较规律即可。

2.7.8　三元运算符

三元运算符可以提供简单的逻辑判断，其应用格式如下：

```
表达式 1?表达式 2:表达式 3
```

如果表达式 1 的值为 TRUE，则执行表达式 2，否则执行表达式 3。

例 2.16　下面举一个简单的例子，来了解三元运算符的使用方法，具体代码如下。（**实例位置：光盘\mr\02\sl\2.16**）

```
<?php
$a="明日科技编程词典";
echo (is_string($a))?$a:"编程无忧"; //使用三元运算符判断变量 a 是否为字符串，是则输出变量 a，否则输出“编程无忧”
?>
```

运行结果如图 2.16 所示。



图 2.16　三元运算符示例运行结果

2.7.9　运算符的使用规则

所谓使用规则就是当表达式中包含多种运算符时，运算符的执行顺序，就相当于数学中的先计算乘除后计算加减一样的道理。PHP 的运算符优先级如表 2.11 所示。

表 2.11　运算符的优先级

优 先 级 别	运　算　符
1	or，and，xor
2	赋值运算符
3	||，&&

续表

优 先 级 别	运　算　符
4	|，^
5	&，.
6	+，-
7	/，*，%
8	<<，>>
9	++，--
10	+，-（正、负号运算符），!，~
11	= =，!=，<>
12	<，<=，>，>=
13	?:
14	->
15	=>

这么多的级别，如果要想都记住是不太现实的，也没有这个必要。如果写的表达式真的很复杂，而且包含了较多的运算符，不妨多加()，例如：

```
$a and (($b != $c) or (5 * (50 – $d)))
```

这样就会减少出现逻辑错误的可能。

2.8　照猫画虎——基本功训练

2.8.1　基本功训练 1——获取当前执行文件名称

视频讲解：光盘\mr\02\lx\获取当前执行文件名称.exe

实例位置：光盘\mr\02\zmhh\01

在 2.4 节中，介绍了很多获取 PHP 信息的常量，其中__FILE__就是获取当前文件名的常量。下面应用该常量获取当前执行文件名称，运行结果如图 2.17 所示。



图 2.17　获取当前执行文件名称

具体步骤如下：

（1）输入 PHP 标记。

```
<?php
?>
```

（2）在标记中定义__FILE__常量，并通过 echo 输出返回值。

```
<?php
echo "当前文件路径为：".__FILE__;                    //使用__FILE__常量获取当前文件路径
?>
```

照猫画虎：通过__LINE__预定义常量获取当前程序的行数。（20 分）（实例位置：光盘\mr\02\zmhh\01_zmhh）

2.8.2　基本功训练 2——加法计算器

视频讲解：光盘\mr\02\lx\加法计算器.exe

实例位置：光盘\mr\02\zmhh\02

下面制作一个简单的加数器，当用户单击“等于”按钮时，通过算术运算符“+”计算两个文本框中数据的和，并显示在文本框中，运行结果如图 2.18 所示。



图 2.18　加法计算器

具体步骤如下：

（1）创建 index.php 文件，设计表单页面，效果如图 2.18 所示。

（2）获取表单中提交的数据，完成加法运算，代码如下：

```
<?php
if(isset($_POST['Submit']) and $_POST['number1']!=null and $_POST['number2']!=null and $_POST['Submit']
=="等于"){
      $number3=$_POST['number1']+$_POST['number2'];                                   //计算变量值
      }else{
      $number3=null;
      }
?>
   <label>
   <input name="number3" type="text" id="number3" size="10" value="<?php echo $number3;?>"/> <!--输出变
量值-->
   </label>
```

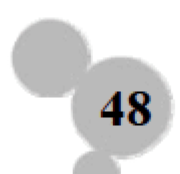
照猫画虎：声明变量 a 的值为 8，变量 b 的值为 9，使用算术运算符“*”来计算变量 a 乘以变量 b 的结果，并使用 echo 语句输出结果。（20 分）（实例位置：光盘\mr\02\zmhh\02_zmhh）

2.8.3　基本功训练 3——输出姚明的个人信息

视频讲解：光盘\mr\02\lx\输出姚明的个人信息.exe

实例位置：光盘\mr\02\zmhh\03

在前面介绍了字符串运算符“.”，该运算符主要用于连接字符串。下面就使用字符串运算符来连接两个字符，输出姚明的个人信息，运行结果如图 2.19 所示。



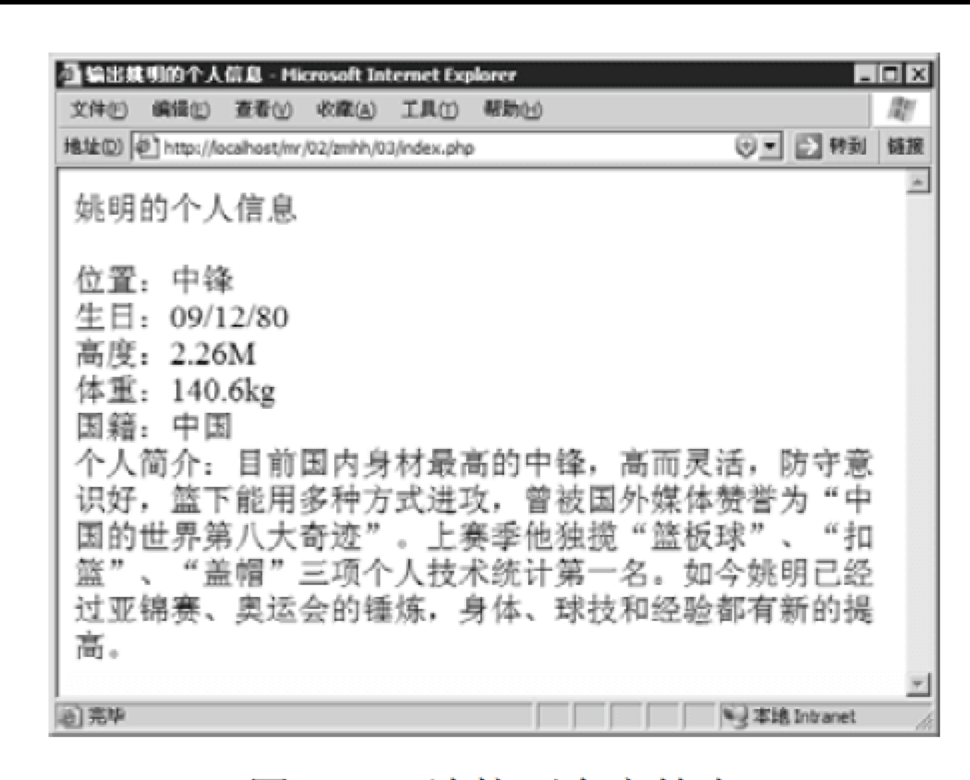


图 2.19　连接两个字符串

具体步骤如下：

（1）输入 PHP 标记，并声明两个变量。

（2）使用字符串运算符“.”连接两个字符串，并使用 echo 语句输出连接后的字符串，具体代码如下：

```
<?php
$a="姚明的个人信息"."<br>"."<br>";
$b="位置：中锋<br>生日：09/12/80<br>高度：2.26M<br>体重：140.6kg<br>国籍：中国<br>个人简介：目前国内身材最高的中锋，高而灵活，防守意识好，篮下能用多种方式进攻，曾被国外媒体赞誉为“中国的世界第八大奇迹”。上赛季他独揽“篮板球”、“扣篮”、“盖帽”三项个人技术统计第一名。如今姚明已经过亚锦赛、奥运会的锤炼，身体、球技和经验都有新的提高。";
echo $a.$b;
?>
```

**照猫画虎：在上面的例子中，在两个字符串中间添加一个换行符，实现字符串的换行输出，换行符符号为（"
"）。（20 分）（实例位置：光盘\mr\02\zmhh\03_zmhh）**

2.8.4　基本功训练 4——比较某一天的产品销量

视频讲解：光盘\mr\02\lx\比较某一天的产品销量.exe

实例位置：光盘\mr\02\zmhh\04

在比较运算符中提供了多种比较方式。本例应用比较运算符中的“<”和“>”来比较两个值的大小，然后使用 var_dump()函数输出比较结果，运行结果如图 2.20 所示。

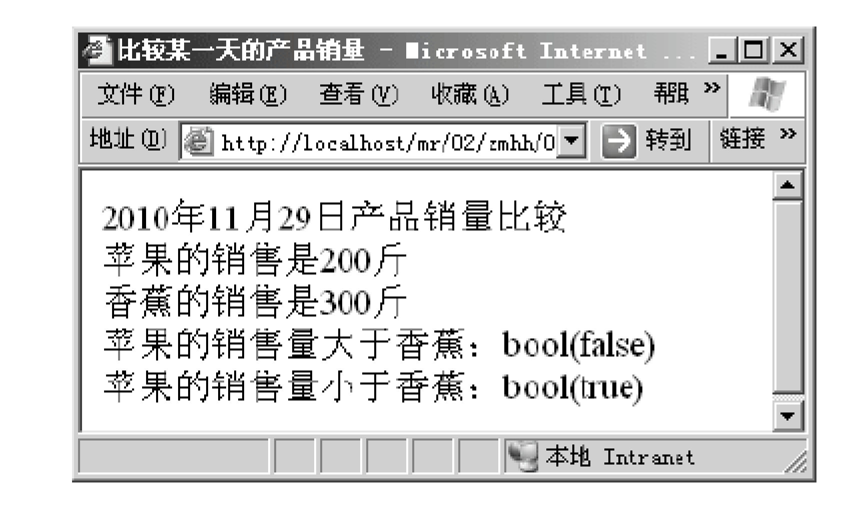


图 2.20　比较两个值的大小

具体步骤如下：

（1）输入 PHP 标记，声明 3 个变量。

（2）使用 echo 语句输出 var_dump()函数判断的返回值，代码如下：

```
<?php
echo "2010 年 11 月 29 日产品销量比较"."<br>";
$a=200;                                  //定义产品销量
$b=300;                                  //定义产品销量
echo "苹果的销售是 200 斤"."<br>";         //输出产品销量
echo "香蕉的销售是 300 斤"."<br>";         //输出产品销量
echo "苹果的销售量大于香蕉：";              //输出字符串文字
```

```
echo var_dump($a>$b)."<br>";                    //输出比较结果值
echo "苹果的销售量小于香蕉：";                    //输出字符串文字
echo var_dump($a<$b)."<br>";                    //输出比较结果值
?>
```

照猫画虎：声明 3 个变量，并分别为其赋值为 6、12、24，然后使用 var_dump()函数来判断 12/6 与 24/12 的结果是否相等，然后使用 echo 语句输出 var_dump()函数的返回值。（20 分）（实例位置：光盘\mr\02\zmhh\04_zmhh）

2.8.5　基本功训练 5——根据工资多少判断购物地点

视频讲解：光盘\mr\02\lx\根据工资多少判断购物地点.exe

实例位置：光盘\mr\02\zmhh\05

从这个训练的名称分析，可以使用三元运算符完成这个判断。这里根据工资的多少为条件，执行不同的操作。判断当工资的值小于 1800 时，执行第二个表达式输出“去远东批发购物”；当工资的值大于 1800 时，执行第 3 个表达式，输出“去国贸商都购物”，运行结果如图 2.21 所示。



图 2.21　一个条件两种答案的判断

具体步骤如下：

（1）编辑 PHP 脚本，定义工资变量。

（2）编写三元运算符，根据工资值进行判断，输出不同的结果。代码如下：

```
<?php
$pay="3000";
echo ($pay<1800)?"去远东批发购物":"去国贸商都购物";
?>
```

照猫画虎：同样应用三元运算符对居民的年龄进行判断，如果年龄大于等于 18，那么输出你是成年人，可以上网吧；否则输出你还未成年，不可以上网吧。（20 分）（实例位置：光盘\mr\02\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

2.9　情景应用——拓展与实践

2.9.1　情景应用 1——在留言板中输出系统当前时间

视频讲解：光盘\mr\02\lx\在留言板中输出系统当前时间.exe

实例位置：光盘\mr\02\qjyy\01

本例中使用时间函数 date 获取当前系统时间，在回复留言页面中输出当前系统时间。运行结果如图 2.22 所示。





图 2.22　输出系统当前时间

具体代码如下：

```
<?php
date_default_timezone_set("Asia/ShangHai");  //将当前时区设置为亚洲上海时区
$date=date('Y/m/d H:i:s');                   //使用 date()函数获取当前系统时间，并将获取到的值赋给$date 变量
echo $date;                                  //输出变量的值
?>
```

DIY：获取当前日期的星期。提示：使用 date 函数中的 w 参数实现。（15 分）（实例位置：光盘\mr\02\qjyy\01_diy）

2.9.2　情景应用 2——根据奇偶天数安排工作任务

视频讲解：光盘\mr\02\lx\根据奇偶天数安排工作任务.exe

实例位置：光盘\mr\02\qjyy\02

奇偶数的区别就在于是否能被 2 整除，下面使用算术运算符和比较运算符来判断变量 a 和变量 b 的奇偶数，运行结果如图 2.23 所示。

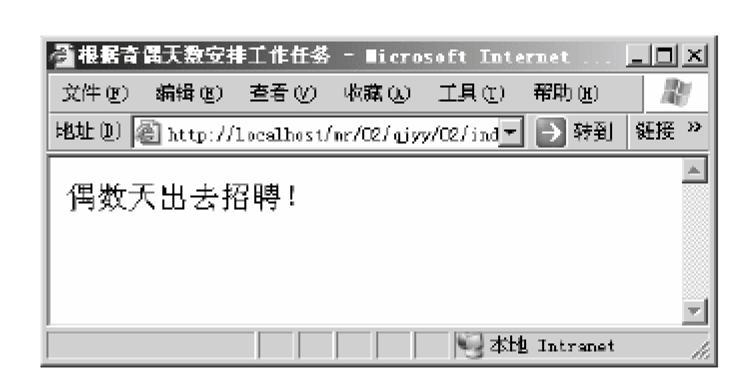


图 2.23　判断奇偶数

本例将工作日期定义为变量，使用运算符来判断奇偶数，然后使用 if…else 语句来判断奇偶天数的工作，具体代码如下：

```
<?php
$a=10;
//使用 if…else 语句判断变量 a 的奇偶数
if($a%2==0){            //判断变量 a 除以 2 后的余数是否为 0，为 0 则证明可以被 2 整除，为偶数；否则为奇数
    echo "偶数天出去招聘！";
}else{
    echo "奇数天在公司做员工培训！";
}
echo "<br>";
?>
```

DIY：判断 2010 年是否为闰年。提示：判断闰年的方法是能被 4 整除但不能被 100 整除，或者该年能被 400 整除则该年为闰年，所以需要使用逻辑运算符。（25 分）（实例位置：光盘\mr\02\qjyy\02_diy）

2.9.3 情景应用 3——计算长方形的面积

视频讲解：光盘\mr\02\lx\计算长方形的面积.exe

实例位置：光盘\mr\02\qjyy\03

计算长方形面积的方法为长乘以宽，本实例中定义了长方形的长和宽，下面通过算术运算符来计算长方形的面积，运行结果如图 2.24 所示。

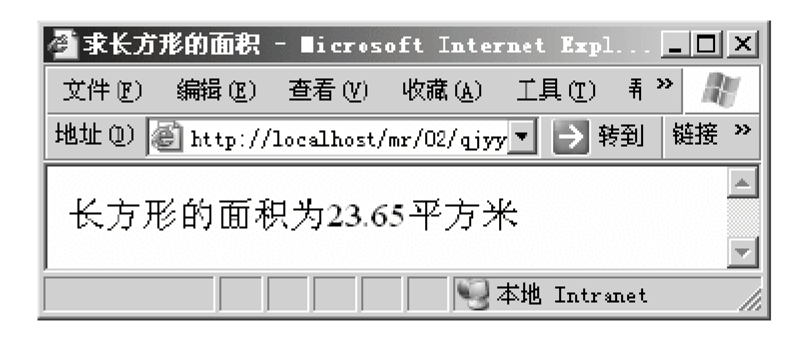


图 2.24 计算长方形的面积

具体代码如下：

```
<?php
$L=4.3;
$H=5.5;
$M=$L*$H;
echo "长方形的面积为".$M."平方米";
?>
```

DIY：同样应用算术运算符，计算楼房的实际使用面积。提示：分别计算每个房间的面积，然后相加得出楼房的实际使用面积。（15 分）（实例位置：光盘\mr\02\qjyy\03_diy）

2.9.4 情景应用 4——检测变量是否为整型

视频讲解：光盘\mr\02\lx\检测变量是否为整型.exe

实例位置：光盘\mr\02\qjyy\04

在 2.6 节中介绍了整型为 int，那么检测变量是否为整型，使用的是 is_int()函数，该函数的语法格式如下：

```
bool is_int(mixed var)
```

参数 var 为要检测的变量。该函数的返回值为布尔型。

下面使用 is_int 函数来检测变量 c 是否为整型，运行结果如图 2.25 所示。



图 2.25 检测变量是否为整型

具体代码如下：

```
<?php
$a=15;
$b=10;
$c=$a/$b;
if(is_int($c)){
	echo "变量 c 是整型";
}else{
	echo "变量 c 不是整型";
```



```
}
?>
```

DIY：根据上面的例子，检测变量 c 是否为浮点型。提示：判断浮点型的函数为 is_float()。（10 分）（实例位置：光盘\mr\02\qjyy\04_diy）

2.9.5　情景应用 5——通过数据递增运算统计网站访问量

视频讲解：光盘\mr\02\lx\通过数据递增运算统计网站访问量.exe

实例位置：光盘\mr\02\qjyy\05

前面已经介绍了递增和递减运算符，下面就应用递增运算符来统计网站访问量，每当网站被浏览一次，访问量就自动增加 1，运行结果如图 2.26 所示。



图 2.26　统计网站访问量

在统计网站访问量的过程中，应用到文件系统函数（fopen 函数、fgets 函数、fclose 函数和 fputs 函数），通过文件系统函数读取文本文件中的数据，并且将更新的数值存储到指定的文本文件中。有关文件系统函数的详细讲解可以参考第 13 堂课的内容，这里只关注其中的递增运算。统计访问量的代码如下：

```
<?php
if(($fp=fopen("counter.txt","r"))==false){
    echo "打开文件失败!";
}else{
    $counter=fgets($fp,1024);                          //读取文件中数据
    fclose($fp);                                       //关闭文本文件
    $counter++;                                        //计数器增加 1
    $fp=fopen("counter.txt","w");                      //以写的方式打开文本文件<!---->
    fputs($fp,$counter);                               //将新的统计数据增加 1
    fclose($fp);
}
if(($fp=fopen("counter.txt","r"))==false){             //打开文件
    echo "打开文件失败!";
}else{
    $counter=fgets($fp,1024);                          //读取文件中数据
    fclose($fp);                                       //关闭文件
```

```
}
?>
```

输出访问量代码如下：

```
<?php echo $counter;?>
```

DIY：根据上面的例子，制作一个限制用户错误登录次数的小程序，限制用户只可以错误登录 10 次，超过 10 次则给出提示。（25 分）（实例位置：光盘\mr\02\qjyy\05_diy）

2.9.6 情景应用 6——输出学生的考试成绩

本例通过字符串运算符“.”连接多个变量，输出李雨翰的考试成绩，运行结果如图 2.27 所示。



图 2.27　连接多个字符串变量

具体代码如下：

```
<?php
$a="语文：98 分";
$b="数学：95 分";
$c="英语：100 分";
$d="排名：第三名";
echo $a."<br>".$b."<br>".$c."<br>".$d;     //使用字符串运算符连接字符串
?>
```

DIY：声明两个字符串变量，分别为“编程词典网”和“www.mrbccd.com”，使用字符串运算符将两个变量连接起来输出。（10 分）（实例位置：光盘\mr\02\qjyy\06_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	6	总分数	
分数								

2.10 自 我 测 试

一、选择题（每题 10 分，5 道题）

1．声明变量时，变量的名称以（　　）符号开头。

A．&　　B．*　　C．$　　D．￥

2．PHP 中的标量数据类型有（　　）种。

A．字符串型（string）　　B．整型（int）　　C．浮点型（float）　　D．布尔型（boolean）

3．不等于运算符是（　　）。

A. ! =　　B. ==　　C. ===　　D. ! ==

4．取余数运算符的符号是（　　）。

A．&　　B．%　　C．¥　　D．#

5．检测数据类型是否为字符串型的函数是（　　）。

A．is_bool　　B．is_string　　C．is_float　　D．is_boolean

二、填空题（每题 10 分，5 道题）

1．使用传值赋值方式为$b 赋值，请将下列代码补充完整。

```
<?php
$a=10;
     (     )
echo $a;
echo $b;
?>
```

2．连接字符串的运算符是（　　）。

3．以下程序的输出结果是（　　）。

```
<?php
$b=201;
    $c=40;
    $a=$b>$c?4:5;
    echo $a;
?>
```

4．执行程序段<?php echo 8%(-2) ?>将输出（　　）。

5．定义常量的函数是（　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

2.11 行动指南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。

续表

序号	内　　容	行 动 指 南	
1	自我测试栏目 分数（　　）	分数>75 分	优秀，有成为编程高手的潜质。
		分数<75 分，	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）声明一个变量，并为变量赋值。	
		（2）声明一个常量，并为常量赋值。	
		（3）声明一个字符串变量，并为其赋值，然后使用函数来检测变量是否为空。	
		（4）声明两个变量，并分别为其赋整型数值，然后使用比较运算符比较两个数值的大小。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

2.12　成功可以复制——杀毒王王江民

成就：我国最早的计算机反病毒专家、江民杀毒软件创始人，拥有各种创造发明 20 多项

创始公司：北京江民新科技术有限公司董事长

财富巅峰：2003 年，入选“中国 IT 富豪榜 50 强”，IT 亿万富翁。

最高学历：初中

出生日期：1951 年 10 月

超级语录：“其实我最喜欢做的还是开发软件，去拯救那些被计算机病毒侵害的计算机。”

“我 38 岁开始学计算机，没有感觉我老了，没有感觉我不行，只感到我的英语基础不好。再说，计算机是实践性非常强的学科。我搞计算机是用计算机，不是学计算机。”

成功历程

王江民，1951 年出生于上海，3 岁因患小儿麻痹后遗症而腿部残疾。初中毕业后，他回到老家山东烟台，从一名街道工厂的学徒工干起，刻苦自学，成长为拥有各种创造发明 20 多项的机械和光电类专家。作为所在国营企业的技术骨干，先后被授予“全国新长征突击手标兵”、“全国青年自学成才标兵”称号。

改变王江民人生的阶段是 1987 年，那时王江民是所在国营企业的技术骨干。当时的国内计算机发展还处于预热期，但经常与高科技打交道的王江民认定计算机必将伴随高科技的发展，改变人类社会的生活。

如果不学，很可能会跟不上科技发展的步伐。“于是先买书，晚上熬夜，每天一点点的看”，开始了一个崭新领域的理论与实践探索之路。王江民说：“我 38 岁开始学计算机，没有感觉我老了，没有感觉我不行，只感到我的英语基础不好。再说，计算机是实践性非常强的学科。我搞计算机是用计算机，不是学计算机。”

机遇偏爱有准备、肯钻研的人，1989 年，王江民花 1000 多元自己买了一台中华学习机，第二年又买了一台 8088PC 机。王江民首先学的是 BASIC 语言。当时，王江民的孩子正上小学一年级，王江民在辅导孩子功课时突发奇想，能否编一程序代替家长辅导。没过多久，3 个月前尚对软件领域一片空白的王江民就编好了一套数学语文教学软件，试过后发现效果奇佳，拿到电脑报参加了一个评奖，结果被誉为“教育软件第一”，并被第一个在全国隆重推广。王江民从一开始就是在用计算机，而不是在学计算机。



软件获奖激发了王江民对软件业的极大兴趣，使他从此把精力转到了软件领域。在开发工控软件时，王江民发现一些或明或暗的病毒。王江民先是用 Debug 手工杀病毒，跟着是写一段程序杀一种病毒。王江民有一个很好的习惯，就是杀一种病毒就在报刊上发表一篇文章，公布这段杀病毒的程序。后来，王江民觉得这些各自独立的杀病毒程序用起来很麻烦，就把 6 个杀不同病毒的程序集成到了一起，命名为 KV6，后来发展到 KV8、KV12、KV18、KV20。随着时间的推移，KV 系列杀毒软件在国内市场中一路高奏凯歌，先后囊括 16 项大奖，成为杀毒软件行业中无可置疑的领头羊。依靠 KV 系列软件，王江民也成为中关村的亿万富翁，并跻身“中国 IT 富豪榜 50 强”，成为新世纪（002280）“知识英雄”的典范，创造了中国软件行业的奇迹。

深度评价

王江民的成功启示我们，每一个想追求成功和传奇的人都会在他身上看到自己的希望和信心，因为王江民各方面的起点都非常低，低到在外人看来凭着王江民的外在条件，他根本就没有任何成功的可能性。后来者们只要少一些浮躁，多一些热情和专注，多一些勤奋和执著，就一定能发现机会，并抓住机遇，书写一段属于你自己的编程传奇。

第 3 堂课

PHP 函数

（视频讲解：100 分钟）

函数分为系统内部函数和用户自定义函数两种。在日常开发中，如果有一个功能或者一段代码要经常使用，就可以把它写成自定义函数，在需要时进行调用。除了自定义函数外，PHP 还提供了庞大的函数库，足有几千种的内置函数，可以直接使用来轻松实现相应功能。在程序中调用函数的目的是为了简化编程的负担、减少代码量和提高效率，达到增加代码重用性、避免重复开发的目的。

学习摘要：

- PHP 函数
- PHP 变量函数库
- PHP 字符串函数库
- PHP 日期时间函数库
- PHP 数学函数库
- PHP 文件系统函数库
- MySQL 函数库

3.1 PHP 函数介绍

在开发过程中，经常要重复某种操作或处理，如数据查询、字符操作等，如果每个模块的操作都要重新输入一次代码，不仅令程序员头痛不已，而且对于代码的后期维护及运行效果也有着较大的影响，使用 PHP 函数即可让这些问题迎刃而解。

3.1.1 定义和调用函数

函数，就是将一些重复使用到的功能写在一个独立的代码块中，在需要时单独调用。创建函数的基本语法格式如下：

```
function fun_name($str1,$stgr2…$strn){
    fun_body;
}
```

参数说明：

☑ function：为声明自定义函数时必须使用到的关键字。

☑ fun_name：为自定义函数的名称。

☑ $str1…$strn：为函数的参数。

☑ fun_body：为自定义函数的主体，是功能实现部分。

当函数被定义后，所要做的就是调用这个函数。调用函数的操作十分简单，只需要引用函数名并赋予正确的参数，即可完成函数的调用。

例 3.01 定义函数 example()，计算传入的参数的平方，然后连同表达式和结果全部输出，代码如下。**（实例位置：光盘\mr\03\sl\3.01）**

```
<?php
    /*    声明自定义函数    */
    function example($num){
        return "$num * $num = ".$num * $num;            //返回计算后的结果
    }
    echo example(10);                                   //调用函数
?>
```

运行结果为：10 * 10 = 100

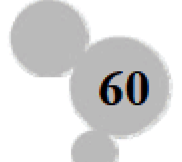
3.1.2 在函数间传递参数

在调用函数时需要向函数传递参数，被传入的参数称为实参，而函数定义的参数为形参。参数传递的方式有按值传递、按引用传递和默认参数 3 种。

1．按值传递方式

按值传递是指将实参的值复制到对应的形参中，在函数内部的操作针对形参进行，操作的结果不会影响到实参，即函数返回后，实参的值不会改变。

例 3.02 首先定义一个函数 example()，功能是将传入的参数值做一些运算后再输出。接着在函数外部定义一个变量$m，也就是要传进来的参数。最后调用函数 example($m)，输出函数的返回值$m 和变量$m 的



值，代码如下。**（实例位置：光盘\mr\03\sl\3.02）**

```
<?php
function example( $m ){                    //定义一个函数
    $m = $m * 5 + 10;
echo "在函数内：\$m = ".$m;                //输出形参的值
}
$m = 1;
example( $m ) ;                            //传递值，将$m 的值传递给形参$m
echo "<p>在函数外  \$m = $m <p>" ;         //实参的值没有发生变化，输出 m=1
?>
```

运行结果如图 3.1 所示。

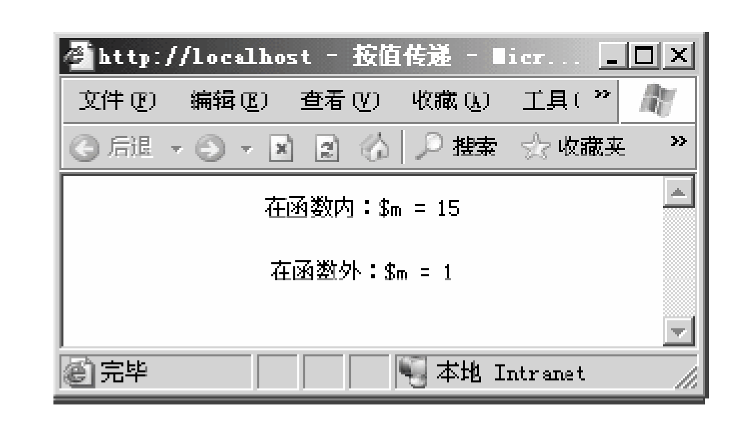


图 3.1　按值传递

2. 按引用传递方式

按引用传递就是将实参的内存地址传递到形参中。这时，在函数内部的所有操作都会影响到实参的值，返回后实参的值会发生变化。引用传递方式就是传值时在原基础上加&号即可。

例 3.03　仍然使用例 3.02 中的代码，唯一不同的地方就是多了一个&号，代码如下。**（实例位置：光盘\mr\03\sl\3.03）**

```
<?php
function example( &$m ){                   //定义一个函数，同时传递参数$m 的变量
    $m = $m * 5 + 10;
echo "在函数内：\$m = ".$m;                //输出形参的值
}
$m = 1;
example( $m ) ;                            //传递值：将$m 的值传递给形参$m
echo "<p>在函数外：\$m = $m <p>" ;         //实参的值发生变化，输出 m=15
?>
```

运行结果如图 3.2 所示。

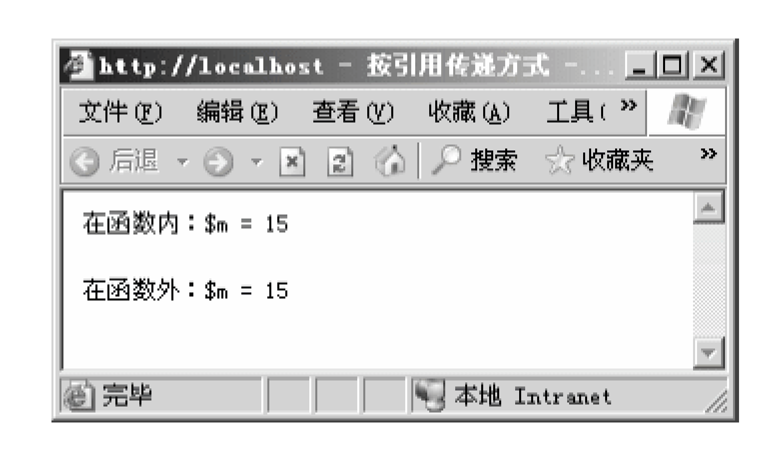


图 3.2　按引用传递方式

3. 默认参数（可选参数）

还有一种设置参数的方式，默认参数即可选参数，可以指定某个参数为可选参数，将可选参数放在参数列表末尾，并且指定其默认值为空。

例 3.04 使用可选参数实现一个简单的价格计算功能，设置自定义函数 values 的参数$tax 为可选参数，其默认值为空。第一次调用该函数，并且给参数$tax 赋值 0.25，输出价格；第二次调用该函数，不给参数$tax 赋值，输出价格，代码如下。**（实例位置：光盘\mr\03\sl\3.04）**

```
<?php
	function values($price,$tax=""){		//定义一个函数，其中的一个参数初始值为空
		$price=$price+($price*$tax);		//声明一个变量$price，等于两个参数的运算结果
		echo "价格:$price<br>";		//输出价格
	}
	values(100,0.25);		//为可选参数赋值 0.25
	values(100);		//没有给可选参数赋值
?>
```

运行结果为：价格:125　　价格:100

注意： 当使用默认参数时，默认参数必须放在非默认参数的右侧，否则函数可能出错。

说明： 从 PHP 5 开始，默认值也可以通过引用传递。

3.1.3　从函数中返回值

前面介绍了如何定义和调用一个函数，并且讲解了如何在函数间传递值，这里将讲解函数的返回值。通常，函数将返回值传递给调用者的方式是使用关键字 return。

return()将函数的值返回给函数的调用者，即将程序控制权返回到调用者的作用域。如果在全局作用域内使用 return()关键字，那么将终止脚本的执行。

例 3.05 使用 return()函数返回一个操作数。先定义函数 values，函数的作用是输入物品的单价、重量，然后计算总金额，最后输出商品的价格，代码如下。**（实例位置：光盘\mr\03\sl\3.05）**

```
<?php
	function values($price,$tax=0.45){		//定义一个函数，函数中的一个参数有默认值
		$price=$price+($price*$tax);		//计算物品金额
		return $price;		//返回金额
	}
	echo values(100);		//调用函数
?>
```

运行结果为：145

return 语句只能返回一个参数，也即只能返回一个值，不能一次返回多个。如果要返回多个结果，就要在函数中定义一个数组，将返回值存储在数组中返回。

3.1.4　变量函数

PHP 支持变量函数。那什么是变量函数？下面通过一个实例来介绍变量函数的具体应用。

例 3.06 首先定义 3 个函数，接着声明一个变量，通过变量来访问不同的函数，代码如下。**（实例位置：光盘\mr\03\sl\3.06）**

```
<?php
function come() {		//定义 come()函数
	echo "来了<p>";
}
```



```
function go($name = "jack") {                //定义 go()函数
    echo " $name 走了<p>";
}
function back($string){                      //定义 back()函数
    echo "又回来了，$string<p>";
}
$func = "come";                              //声明一个变量，将变量赋值为“come”
$func();                                     //使用变量函数来调用函数 come()
$func = "go";                                //重新给变量赋值
$func("Tom");                                //使用变量函数来调用函数 go()
$func = "back";                              //重新给变量赋值
$func("Lily");                               //使用变量函数来调用函数 back()
?>
```

运行结果如图 3.3 所示。



图 3.3　变量函数

可以看到，函数的调用是通过改变变量名来实现的，通过在变量名后面加上一对小括号，PHP 将自动寻找与变量名相同的函数，并且执行它。如果找不到对应的函数，系统将会报错。

3.1.5　对函数的引用

按引用传递参数可以修改实参的内容。引用不仅可用于普通变量、函数参数，也可用于函数本身。对函数的引用，就是对函数返回结果的引用。

例 3.07　首先定义一个函数，在函数名前加“&”符。接着通过变量$str 引用该函数，最后输出变量$str，实际上就是$tmp 的值，代码如下。（**实例位置：光盘\mr\03\sl\3.07**）

```
<?php
function &example($tmp=0){                   //定义一个函数，注意加“&”符
     return $tmp;                            //返回参数$tmp
}
$str = &example("看到了");                   //声明一个函数的引用$str;
echo $str."<p>";                             //输出$str
?>
```

运行结果为：看到了

注意：和参数传递不同，这里必须在两个地方使用“&”符，用来说明返回的是一个引用。

3.1.6　取消引用

当不再需要引用时，可以取消引用。取消引用使用 unset()函数，它只是断开了变量名和变量内容之间

的绑定，而不是销毁变量内容。

例 3.08 首先声明一个变量和变量的引用，输出引用后取消引用，再次调用引用和原变量。可以看到，取消引用后对原变量没有任何影响，代码如下。**（实例位置：光盘\mr\03\sl\3.08）**

```
<?php
    $num = 1234;                          //声明一个整型变量
    $math = &$num;                        //声明一个对变量$num 的引用$math
    echo "\$math is:  ".$math."<br>";     //输出引用$math
    unset($math);                         //取消引用$math
    echo "\$math is: ".$math."<br>";      //再次输出引用
    echo "\$num is:  ".$num;              //输出原变量
?>
```

运行结果如图 3.4 所示。



图 3.4 取消引用

3.2 PHP 变量函数库

除了用户自行编写的函数库外，PHP 自身也提供了很多内置的函数，PHP 变量函数库就是其中一个。但并不是所有的函数都会经常用到。因此，读者只要熟悉一些常用的函数即可。本堂课将面向实际开发的需求，向读者介绍一些常用的变量函数，如表 3.1 所示。

表 3.1 常用的变量函数介绍

类 型	说 明
empty	检查一个变量是否为空，为空，返回 TRUE；否则返回 FALSE
gettype	获取变量的类型
intval	获取变量的整数值
is_array	检查变量是否为数组类型
is_int	检查变量是否为整数
is_numeric	检查变量是否为数字或由数字组成的字符串
isset	检查变量是否被设置，即是否被赋值
print_r	打印变量
settype	设置变量的类型，可将变量设为另一个类型
unset	释放给定的变量，即销毁这个变量
var_dump	打印变量的相关信息

以上是一些常用变量函数及其功能的简单介绍。其他变量函数的具体介绍请参见 PHP 手册。

isset()函数检查变量是否被设置，设置则返回 TRUE，否则返回 FALSE。该函数的语法如下：



```
bool isset ( mixed var [, mixed var [, ...]])
```

参数说明：第一个 var 为必要参数，是输入的变量。第二个 var 为可选参数，是输入的变量，可有多个。

例 3.09　应用 isset()函数检测变量是否被设置，代码如下。**（实例位置：光盘\mr\03\sl\3.09）**

```
<?php
$var = '';
if (isset($var)) {
    echo "变量被设置 1<br />";
}
unset ($var);                          //销毁变量
if (isset($var)) {
    echo "变量被设置 2<br />";
}
$var = NULL;                           //将变量设置为 null
if (isset($var)) {
    echo "变量被设置 3<br />";
}
?>
```

运行结果为：变量被设置 1

注意： isset()只能用于变量，因为传递任何其他参数都将造成解析错误。若想检测常量是否已设置，可使用 defined()函数。

3.3　PHP 字符串函数库

PHP 字符串函数库是 PHP 开发中一项非常重要的内容，必须掌握其中常用函数的使用方法。表 3.2 对 PHP 常用的字符串函数进行了总结。

表 3.2　常用字符串介绍

函　　数	功　　能
addcslashes	实现转义字符串中的字符，即在指定的字符前面加上反斜线
explode	将字符串依指定的字符串或字符 separator 切开
echo	用来输出字符串
ltrim	删除字符串开头的连续空白
md5	计算字符串的 md5 哈希
strlen	获取指定字符串的长度
str_ireplace	将某个指定的字符串都替换为另一个指定的字符串（大小写不敏感）
str_repeat	将指定的字符串重复输出
str_replace	取代所有在字串中出现的字串
strchr	获取指定字符串（A）在另一个字符串（B）中首次出现的位置
stristr	获取指定字符串（A）在另一个字符串（B）中首次出现的位置到（B）字符串末尾的所有字符串
strstr	获取一个指定字符串在另一个字符串中首次出现的位置到后者末尾的子字符串
substr_replace	将字符串中的部分字符串替换为指定的字符串
substr	从指定的字符串 str 中按照指定的位置 start 截取一定长度 length 的字符

1．explode()函数

explode()函数将字符串依指定的字符串或字符 separator 切开。该函数的语法如下：

```
array explode(string separator, string string, [int limit])
```

返回由字符串组成的数组，每个元素都是 string 的一个子串，它们被字符串 separator 作为边界点分隔出来。如果设置了 limit 参数，则返回的数组包含最多 limit 个元素，而最后那个元素将包含 string 的剩余部分；如果 separator 为空字符串（""），explode()函数将返回 FALSE；如果 separator 所包含的值在 string 中找不到，那么 explode()函数将返回包含 string 单个元素的数组；如果参数 limit 是负数，则返回除了最后的-limit 个元素外的所有元素。

例 3.10 应用 explode()函数来获取变量中的字符串，其中以"："进行分隔，代码如下。**（实例位置：光盘\mr\03\sl\3.10）**

```
<?php
$data = "用户名:明日科技:密码:12346**:电话:12345678";    //声明一个变量
list($user,$users ,$pass,$passs ,$tel, $tels) = explode(":", $data);   //使用 explode()函数分割字符串，并输出数组中元素
echo $users . "<br>";       //输出变量的值
echo $passs . "<br>";       //输出变量的值
echo $tels;
?>
```

运行结果为：明日科技　　12346**　　12345678

你问我答：explode()函数在实际中有哪些应用？

在投票系统的后台管理中，经常需要添加投票选项到投票系统中去，作为投票的内容。这里在添加投票选项时就应用 explode()函数对添加的投票选项进行区分，然后分别在页面中输出添加的内容。

2．md5()函数

md5 函数计算字符串的 md5 哈希，该函数是一种编码的方式，但是不能解码。该函数的语法如下：

```
string md5 ( string str , bool raw_output )
```

参数 str 为被加密的字符串；参数 raw_output 为布尔型，TRUE 表示加密字符串以二进制格式返回。

例如，应用 md5()函数对字符串"明日科技"进行编码，代码如下：

```
<?php
     echo md5("明日科技");   //63bac9345fcfdf1ec8cfa82f1c996c29
?>
```

运行结果为：63bac9345fcfdf1ec8cfa82f1c996c29

说明：通常为了保证用户注册信息的安全，用 md5()函数对用户注册的密码进行加密。

3.4　PHP 日期时间函数库

PHP 通过内置的日期和时间函数，完成对日期和时间的各种操作，常用的日期和时间函数如表 3.3 所示。



表 3.3　常用的日期和时间函数

函　　数	功　　能
checkdate	验证日期的有效性
date	格式化一个本地时间/日期
microtime	返回当前 UNIX 时间戳和微秒数
mktime	获取一个日期的 UNIX 时间戳
strftime	根据区域设置格式化本地时间/日期
strtotime	将任何英文文本的日期时间描述解析为 UNIX 时间戳
time	返回当前的 UNIX 时间戳

1．checkdate()函数

checkdate()函数用于验证日期的有效性，如果日期有效则返回 TRUE，否则返回 FALSE。该函数的语法如下：

```
bool checkdate ( int month, int day, int year)
```

参数说明：month 的有效值是从 1 到 12；day 的有效值在给定的 month 所应该具有的天数范围之内，包括闰年；year 的有效值是从 1 到 32767。

例如，应用 checkdate()函数判断日期是否有效，如果正确则输出 1，否则不输出。代码如下：

```
<?php
$checkdate=checkdate(7,32,2008);          //判断日期是否有效
echo $checkdate;                          //输出变量
?>
```

运行结果为：1

2．mktime()函数

mktime()函数用于返回一个日期的 UNIX 时间戳。该函数的语法如下：

```
int mktime ( int hour, int minute, int second ,int month, int day ,int year , int is_dst)
```

mktime()函数根据给出的参数返回 UNIX 时间戳。时间戳是一个长整数，包含了从 UNIX 新纪元（1970 年 1 月 1 日）到给定时间的秒数。其参数可以从右向左省略，任何省略的参数会被设置成本地日期和时间的当前值。其中参数 is_dst 在夏令时可以被设为 1，如果不是则设为 0；如果不知道是否为夏令时则设为−1（默认值）。

例 3.11　应用 mktime()函数获取系统当前的时间戳，然后通过 date()函数来对其进行格式化，输出时间，代码如下。**（实例位置：光盘\mr\03\sl\3.11）**

```
<?php
echo "mktime 函数返回的时间戳:". mktime()."<br>";
echo date( "M-d-Y", mktime());
?>
```

运行结果为：mktime()函数返回的时间戳:1290063618　　Nov-18-2010

注意：由于运行程序的日期不同，所以获取的时间戳和日期不同。

3.5　PHP 数学函数库

PHP 提供了大量的内置数学函数，大大提高了开发人员在数学运算上的精准度。表 3.4 介绍了一些常用

的 PHP 数学函数。

表 3.4　常用的 PHP 数学函数

类　型	说　明
ceil	返回不小于参数 value 值的最小整数，如果有小数部分则进一位
mt_rand	返回随机数中的一个值
mt_srand	配置随机数的种子
rand	产生一个随机数，返回随机数的值
round	实现对浮点数进行四舍五入
floor	实现舍去法取整，该函数返回不大于参数 value 值的下一个整数，将 value 值的小数部分舍去取整
fmod	返回除法的浮点数余数
getrandmax	获取随机数最大的可能值
max	返回参数中的最大值
min	返回参数中的最小值

1．floor()函数

floor()函数实现舍去法取整，返回不大于参数 value 值的下一个整数，将 value 值的小数部分舍去取整。floor()函数返回的类型是浮点型，因为浮点型值的范围通常比整型要大。该函数的语法如下：

```
float floor ( float value)
```

例如，应用 floor()函数对数值 6、7.2 和 8.9 进行取整，代码如下：

```
<?php
echo floor(6). "<br>";
echo floor(7.2). "<br>";
echo floor(8.9). "<br>";
?>
```

运行结果为：6　7　 8

2．fmod()函数

fmod()函数返回除法的浮点数余数。该函数的语法如下：

```
float fmod ( float x, float y)
```

该函数返回被除数（x）除以除数（y）所得的浮点数余数。

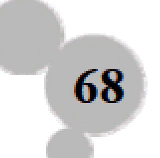
例如，应用 fmod()函数获取 5 除以 1.5 所得的余数，代码如下：

```
<?php
$x = 5;
$y = 1.5;
$z = fmod($x, $y);
echo $z;
?>
```

运行结果为：0.5

3.6　PHP 文件系统函数库

文件是存取数据的方式之一。相对于数据库来说，文件在使用上更方便、直接。如果数据较少、较简



单，使用文件无疑是最合适的方法。PHP 对文件的操作是通过内置的文件操作系统函数来完成的。常用的文件系统操作函数如表 3.5 所示。

表 3.5 常用的文件系统函数

函 数	功 能
basename	返回文件路径中基本的文件名
copy	将某文件由当前目录复制到其他目录，如果成功则返回 TRUE，否则返回 FALSE
file_exists	判断指定的目录或文件是否存在。如果存在则返回 TRUE，否则返回 FALSE
file_put_contents	将字符串写入到指定的文件中
file	读取某文件的内容，并将结果保存到数组中，数组内每个元素的内容对应读取文件的一行
filetype	返回文件的类型。可能的值有 fifo、char、dir、block、link、file 和 unknown
fopen	打开某文件，并返回该文件的标识指针。该文件可以是本地的，也可以是远程的
fread	从文件指针所指文件中读取指定长度的数据
is_dir	如果该函数参数所代表的路径为目录并且该目录存在，则返回 TRUE，否则返回 FALSE
is_uploaded_file	判断文件是否应用 HTTP POST 方式上传的，如果是则返回 TRUE，否则返回 FALSE
mkdir	新建一个目录
move_uploaded_file	应用 POST 方法上传文件
readfile	读入一个文件，并将读入的内容写入到输出缓冲
rmdir	删除指定的目录，如果删除成功则返回 TRUE，否则返回 FALSE
unlink	用于删除文件，如果删除成功则返回 TRUE，否则返回 FALSE

1．fopen()函数

fopen()函数用于打开某文件，并返回该文件的标识指针。该文件可以是本地的，也可以是远程的。fopen()函数的语法如下：

```
resource fopen(string filename, string mode [, int use_include_path [, resource context]])
```

fopen ()函数的参数说明如表 3.6 所示。

表 3.6 fopen()函数的参数说明

参 数	说 明
filename	必要参数。用于指定要打开文件的本地或远程地址
mode	必要参数。用于指定要打开文件的模式
use_include_path	可选参数。如果将该参数设置为 TRUE，PHP 会尝试按照 include_path 标准包含路径中的每个指向去打开文件
context	可选参数。设置提高文件流性能的一些选项

2．mkdir()函数

mkdir()函数用于判断某文件是否存在，并且是否可写。如果存在，并且可写则返回 TRUE；否则返回 FALSE。mkdir()函数的语法如下：

```
bool mkdir(string pathname [, int mode])
```

参数说明：pathname 为必要参数，用于指定新建目录的名称；mode 为可选参数，用于指定新建目录的模式。

例 3.12 下面就应用文件和目录的操作技术编写一个实例，实现文件、目录的创建和删除功能，代码

如下。**（实例位置：光盘\mr\03\sl\3.12）**

创建 index.php 文件，首先判断文件夹和文件是否存在，如果存在则利用 file_get_contents()函数读取文本文件信息，然后删除文件夹及文件；如果不存在则使用 mkdir()和 fopen()函数创建文件夹和文件并向文件中写入文本信息。代码如下：

```
<?php
    if(!is_dir('txt')){                                         //判断 txt 是不是文件夹目录
        mkdir('txt');                                           //创建名为 txt 的文件夹目录
        $open=fopen("txt/in.txt","w+");                         //以读写方式打开文件 in.txt
            if(is_writable("txt/in.txt")){                      //如果此文件为可写模式
                if(fwrite($open,"今天是美好的一天，一定要开心哦！")>0){ //当文件写入成功时
                    fclose($open);                              //关闭文件
                    echo "<script>alert('写入成功')</script>";              //输出成功提示
                }
            }
    }else{                                                      //如果 txt 文件目录存在
        if(is_file("txt/in.txt")){                              //判断目录下是否存在 in.txt 文件
            if(is_readable("txt/in.txt")){                      //判断此文件是否可读
                echo file_get_contents("txt/in.txt");           //输出文本数据信息
                unlink("txt/in.txt");                           //删除 in.txt 文件
                rmdir('txt');                                   //删除目录
            }
        }
    }
?>
```

运行结果如图 3.5 和图 3.6 所示。



图 3.5　输出文件信息及删除文件和文件夹



图 3.6　创建文件夹及文件并写入信息

3.7　MySQL 函数库

PHP 支持多种数据库，而且内置了很多数据库操作的函数库，其中 MySQL 函数库是最为常用的一种。表 3.7 列举了 MySQL 函数库中的常用函数。

表 3.7　常用的 MySQL 函数

函　　数	功　　能
mysql_close	关闭 MySQL 连接
mysql_connect	打开一个到 MySQL 服务器的连接
mysql_create_db	新建一个 MySQL 数据库
mysql_error	返回上一个 MySQL 操作产生的文本错误信息
mysql_fetch_array	从结果集中获取一行作为关联数组，或数字数组，或两者兼有

续表

函　　数	功　　能
mysql_fetch_assoc	从结果集中获取一行作为关联数组
mysql_fetch_field	从结果集中获取列信息并作为对象返回
mysql_fetch_object	从结果集中获取一行作为对象
mysql_fetch_row	从结果集中获取一行作为枚举数组
mysql_num_rows	获取结果集中行的数目
mysql_query	发送一条 MySQL 查询
mysql_select_db	选择 MySQL 数据库

说明： 这里只是对 MySQL 函数库中的常用函数做了简单的介绍，有关 MySQL 函数库中函数的详细讲解读者可以参考本书中的第 8 堂课。

3.8　照猫画虎——基本功训练

3.8.1　基本功训练 1——判断用户提交用户名是否为空

视频讲解：光盘\mr\03\lx\判断用户提交用户名是否为空.exe

实例位置：光盘\mr\03\zmhh\01

判断用户提交的用户名是否为空应用 empty()函数，如果变量是非空或非零的值，empty()函数返回 FALSE；否则，当变量为""、0、"0"、null、FALSE、array()、var $var，以及没有任何属性的对象都将被认为是空的，返回 TRUE。

本实例应用 empty()函数判断用户提交的用户名或者密码是否为空，代码如下：

```
<?php
if(isset($_POST['Submit']) && $_POST['Submit']=="登录"){          //通过 isset()函数对登录按钮进行判断
$user=$_POST['user'];                                              //通过$_POST 函数调用表单文本域的值
$pass=$_POST['pass'];
	if(empty($user)||empty($pass)){                                 //通过 if 语句判断用户名或密码不能为空
		echo "<script>alert('用户名或密码不能为空');</script>";     //用户名或密码为空时，给出提示
	}else{
		echo"输入的用户名为：$user  密码为：$pass";                  //不为空时，输出用户名与密码
	}
}
?>
```

运行结果如图 3.7 所示。

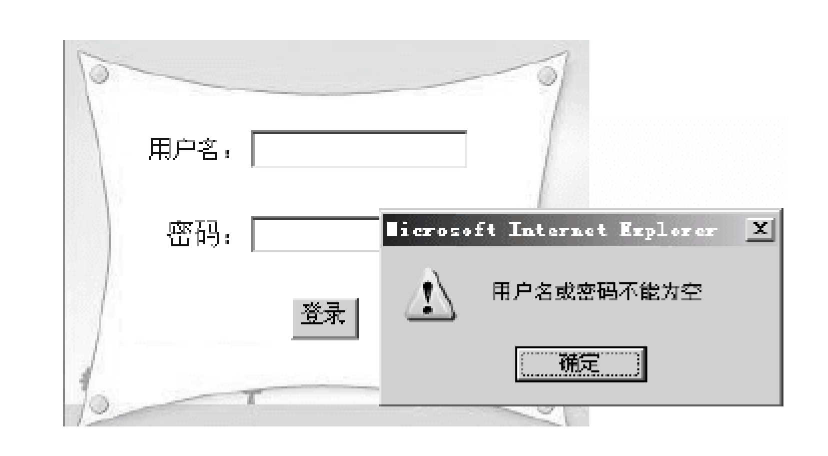


图 3.7　用户名或密码不能为空

照猫画虎：输出用户名和密码。（8 分）（实例位置：光盘\mr\03\zmhh\01_zmhh）

3.8.2　基本功训练 2——打印输出员工的个人信息

视频讲解：光盘\mr\03\lx\打印输出员工的个人信息.exe

实例位置：光盘\mr\03\zmhh\02

var_dump()函数可以显示关于一个或多个表达式的结构信息，包括表达式的类型与值。数组将递归展开值，通过缩进显示其结构。本实例应用 var_dump()函数列出数组的结构信息。代码如下：

```
<?php
/*声明一个数组，数组内容为要打印输出的元素*/
$array = array("a"=>"姓名：李**"."<br>","p"=>"性别：女"."<br>","学历：本科"."<br>","年龄：25");
var_dump($array);    //应用 var_dump()函数输出数组元素
?>
```

运行结果如图 3.8 所示。

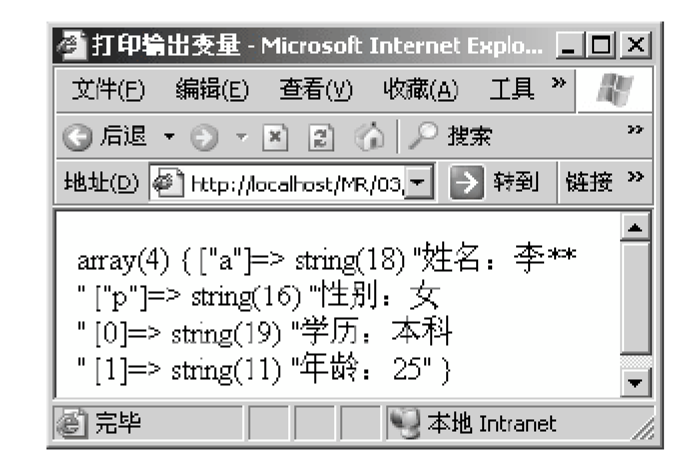


图 3.8　var_dump()打印输出数组元素

照猫画虎：应用 print_r()函数输出上面例子中的数组元素。（12 分）（实例位置：光盘\mr\03\zmhh\02_zmhh）

3.8.3　基本功训练 3——输出公司的企业文化

视频讲解：光盘\mr\03\lx\输出公司的企业文化.exe

实例位置：光盘\mr\03\zmhh\03

本实例应用 echo()函数输出公司的企业文化，代码如下：

```
<?php
        echo"
            经营理念：经营用户比单纯争夺用户更重要
             <br>
          企业道德：让用户花更少的钱，获得更多、更专业、更有价值的内容和服务";
?>
```

运行结果如图 3.9 所示。

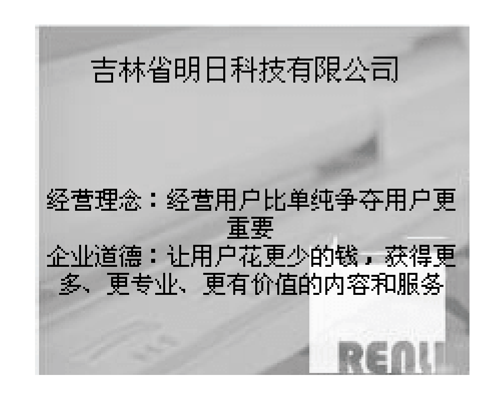


图 3.9　应用 echo()输出公司的企业文化

注意：echo 只是用来输出字符串。由于它不是真正的函数，因此没有返回值。

照猫画虎：将指定的字符串重复输出。提示：可以用 str_repeat()函数实现。（20 分）（实例位置：光盘\mr\03\zmhh\03_zmhh）

3.8.4　基本功训练 4——计算密码长度

视频讲解：光盘\mr\03\lx\计算密码长度.exe

实例位置：光盘\mr\03\zmhh\04

对密码的长度进行计算，以及对密码长度进行限制，在实际应用中比较广泛，尤其是实现会员注册功能。本实例介绍一种应用 strlen()函数计算输入的密码长度的方法，运行结果如图 3.10 所示。



图 3.10　计算密码长度显示界面

具体步骤如下：

（1）通过 Dreamweaver 设计一个 form 表单，表单元素为一个提交密码的文本域，一个“计算”按钮。代码如下：

```
<form action="" method="post" name="form1" class="STYLE1" id="form1">
<td width="411" bgcolor="#99CC33"><div align="center"><span class="STYLE2">密码：
            <input name="pass" type="password" size="12" />
       </span>
              <input type="submit" name="Submit" value="计算" />
</form>
```

（2）在本页中通过 isset()函数判断用户提交的密码和按钮值是否存在。如果存在则通过$_POST[]全局数组获取表单中提交的密码，通过 strlen()函数计算出密码的长度，并在页面上显示。代码如下：

```
<?php
$pass="";
if(isset($_POST['pass'])&&$_POST['pass']!=""){          //对 form 表单的文本域进行判断
$pass=$_POST['pass'];
      echo strlen($pass);                               //通过 strlen()函数计算文本域中的密码长度
}
?>
```

照猫画虎：对密码长度进行限制（提示：通过 strlen()函数来控制密码的长度）。（20 分）（实例位置：光盘\mr\03\zmhh\04_zmhh）

3.8.5　基本功训练 5——验证输入时间是否有效

视频讲解：光盘\mr\03\lx\验证输入时间是否有效.exe

实例位置：光盘\mr\03\zmhh\05

一年有 12 个月，一个月有 31 天（或 30 天，2 月 28 天，闰年为 29 天），一星期有 7 天……这些常识都是人人皆知的事。但计算机并不能自己分辨数据的对与错，只是依靠开发者提供的功能去执行或检查。PHP 中内置了日期检查函数，即 checkdate()函数。本实例应用 checkdate()函数判断在数据录入系统中输入的日期是否有效，如果有效，则输出数据录入成功；否则，将弹出提示对话框，运行结果如图 3.11 所示。



图 3.11　验证日期是否有效

具体步骤如下：

首先，创建一个 form 表单，添加 5 个表单元素，分别是产品名称、生产日期和提交按钮。关键代码如下：

```
<form id="form1" name="form1" method="post" action="">
</tr>
  <tr align="right">
    <td height="30">生产日期：</td>
    <td align="left" bgcolor="#009999"><input name="textfield2" type="text" size="4" />年
<input name="month" type="text" id="month" size="4" />月
<input name="day" type="text" id="day" size="4" />日 </td>
  </tr>
  <tr>
    <td height="30" colspan="2" align="center"><input type="submit" name="Submit" value="提交" /></td>
  </tr>
  </form>
```

然后，在本页中应用 isset()函数对表单提交的生产日期进行判断，如果存在则通过 checkdate()函数对生产日期进行验证，并且输出验证结果。关键代码如下：

```
<?php
if(isset($_POST['Submit']) && $_POST['Submit']!=""){ //对 form 的提交按钮进行判断，调用日期文本域提交的内容
     $year=$_POST['year'];
     $month=$_POST['month'];
     $day=$_POST['day'];
     if(checkdate($month,$day,$year)==TRUE){     //通过 checkdate()函数验证文本域提交的日期是否有效
          echo "数据录入成功!!";
     }else{
          echo "<script> alert('您输入的日期不合法!!');</script>";
     }
}
?>
```

照猫画虎：在上述实例基础上，将通过验证的合法日期进行输出。（20 分）（实例位置：光盘\mr\03\zmhh\05_zmhh）

3.8.6　基本功训练 6——获取商品销量的最高值

视频讲解：光盘\mr\03\lx\获取商品销量的最高值.exe

实例位置：光盘\mr\03\zmhh\06

想要获取数字中的最大值，可以使用 max()函数。max()函数被广泛应用于网站的最高访问量的统计或



者销售系统中的销售统计。在本实例中，应用 max()函数获取一年中商品月销量最高的值，运行结果如图 3.12 所示。



图 3.12　应用 max()函数获取月销量最高值

具体步骤如下：

（1）创建一个 form 表单，添加表单元素，用于提交一年中每个月的销售数据。

（2）在本页中编写 PHP 脚本，通过 isset()函数判断是否执行了提交操作。如果执行提交操作，那么将 $_POST[]全局数组获取的每个月的销售数据赋给指定的变量，并且将这些数据存储到一个数组中，最终通过 max()函数统计数组中的最大值；如果没有执行提交操作，那么为执行的变量赋空值。关键代码如下：

```
<?php
if(isset($_POST['Submit']) && $_POST['Submit']=="提交"){
      $month1=$_POST['month1'];
      $month2=$_POST['month2'];
//此处省略了部分代码
$array=array($_POST['month1'],$_POST['month2'],$_POST['month3'],$_POST['month4'],$_POST ['month5'],$_ POST ['month6'], $_POST['month7'],$_POST['month8'],$_POST['month9'],$_ POST['month10'], $_POST['month11'], $_POST ['month12']);
        $max=max($array);
 }else{
      $month1="";
      $month2="";
//此处省略了部分代码
      $max="";
 }
?>
```

（3）输出月销售量最高的数据。

照猫画虎：获取一年中销量最低值。提示：应用 min()函数实现。（10 分）（实例位置：光盘\mr\03\zmhh\06_zmhh）

3.8.7　基本功训练 7——读取文本文件中的数据

视频讲解：光盘\mr\03\lx\读取文本文件中的数据.exe

实例位置：光盘\mr\03\zmhh\07

读取文件中的数据，可以读取一个字符、一行字串或整个文件，还可以读取任意长度的字串。在本实例中应用 fread()函数读取整个文本文件中的数据，运行结果如图 3.13 所示。

具体步骤如下：

（1）定义文本文件在实例根目录下的存储位置。

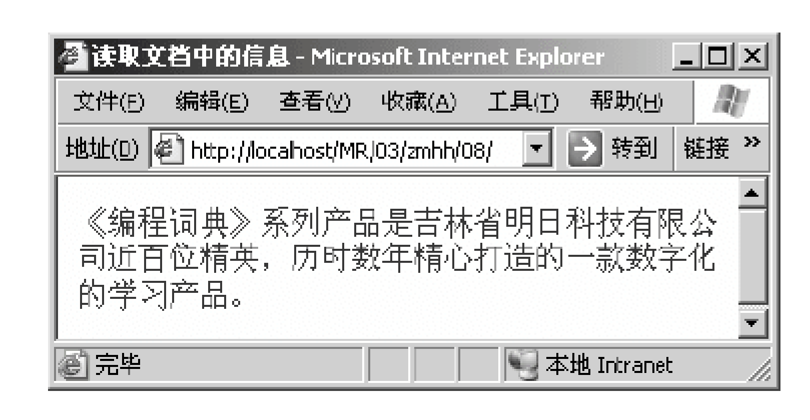


图 3.13　fread()函数读取文本文件中数据

（2）利用 fopen()函数以只读方式打开文件并返回文件句柄。

（3）利用 filesize()函数获取文本文件数据的长度。

（4）利用 fread()函数输出文本数据。代码如下：

```
<?php
      $path="txt/in.txt";                     //定义文本文件路径
      $open=fopen($path,"r");                 //打开指定文件
      $size=filesize($path);                  //获取文本数据长度
      echo fread($open,$size);                //输出所有数据
      fclose($open);                          //关闭文件
?>
```

照猫画虎：在文本文件中写入“我想学 PHP”。提示：应用 fwrite()函数写入。（10 分）（实例位置：光盘\mr\03\zmhh\07_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	6	7	总分数	
分数									

3.9　情景应用——拓展与实践

3.9.1　情景应用 1——控制帖子标题的输出长度

视频讲解：光盘\mr\03\lx\控制帖子标题的输出长度.exe

实例位置：光盘\mr\03\qjyy\01

在社区或者论坛发表帖子时，当输入帖子的标题时，经常会看到限制字数的提示。一方面是为了保证版面的规整，另一方面是方便管理。在本实例中通过 strlen()函数来控制帖子标题的输出长度，运行结果如图 3.14 所示。

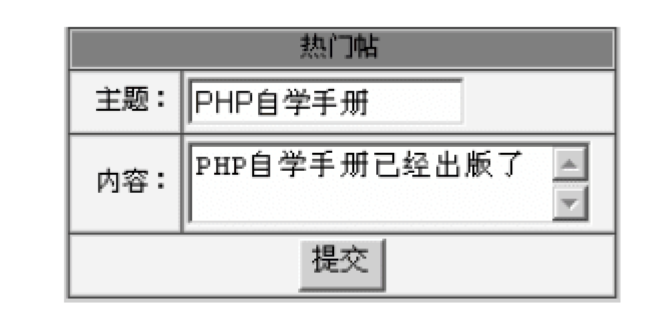


图 3.14　控制帖子的输出长度

具体步骤如下：

首先，创建一个 form 表单，添加 3 个表单元素：主题的文本域和发表内容的文本区域，以及一个提交按钮。关键代码如下：

```
<form id="form1" name="form1" method="post" action="">
<!--省略部分为表单元素代码-->
<td width="46"><div align="right">主题：</div></td>
<td width="190"><input name="zhuti" type="text" size="15" /></td>
<!--省略部分为表单元素代码-->
<td><div align="right">内容：</div></td>
```

```
<td><textarea name="textarea"></textarea></td>
<!--省略部分为表单元素代码-->
<td colspan="2"><div align="center"><input type="submit" name="Submit" value="提交" /></td>
</form>
```

然后，在本页中编写 PHP 脚本，通过 isset()函数判断主题文本域的值是否为空，如果不为空，则通过$_POST[]全局数组获取表单中提交的标题，并且通过 strlen()函数判断标题的长度是否超过指定的范围，最后输出提示信息。关键代码如下：

```
<?php
if(isset($_POST['Submit']) && $_POST['Submit']=="提交"){        //对提交按钮进行判断
        if(isset($_POST['zhuti']) && $_POST['zhuti']!="" ){      //对文本框进行判断，当文本框非空时，执行下面的操作
                $zt=$_POST['zhuti'];                              //调用表单文本域提交的内容
                $str=strlen($zt);                                 //通过函数 strlen()对标题长度进行判断
                if($str<=15){                                     //当输入的字符数小于 15 时，执行下面的操作
                        echo "您输入的标题可用";
                }else{
                        echo "您输入的标题必须在 15 个字节以内";
                }
        }else{
                echo "<script>alert('提交内容不能为空');</script>";
        }
}
?>
```

说明：由于采用的编码格式不同，在限制输出的字符个数上有差别，本实例的编码格式为 UTF-8（一个中文字符相当于 3 个英文字符）。

DIY：判断帖子主题中是否出现特殊字符。提示：应用 substr_count()完成。（10 分）（实例位置：光盘\mr\03\qjyy\01_diy）

3.9.2　情景应用 2——去除用户填写注册信息中的空格

视频讲解：光盘\mr\03\lx\去除用户填写注册信息中的空格.exe

实例位置：光盘\mr\03\qjyy\02

用户在输入数据时，经常会在无意中输入多余的空白字符，在某些情况下，字符串中不允许出现空白字符和特殊字符，这就需要去除字符串中的空白字符和特殊字符。在 PHP 中提供 trim()函数去除字符串左右两边的空白字符和特殊字符、ltrim()函数去除字符串左边的空白字符和特殊字符、rtrim()函数去除字符串中右边的空白字符和特殊字符。在本实例中应用 ltrim()函数去除用户名中左侧的空格，如图 3.15 所示。未去除空格的效果如图 3.16 所示。

填写注册信息	
用户名：	大麦
密码：	●●●●
电话：	123456
QQ：	15564
地址：	吉林省长春市
	提交

图 3.15　去除空格后的注册表

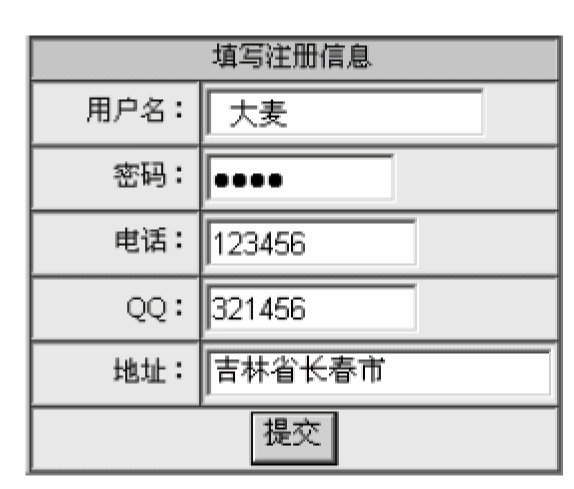


图 3.16　未去除空格的注册表

去除注册用户名左侧空格的关键代码如下：

```
<?php
$strs="";                                                   //声明关于文本域的变量
$pass="";
$tel="";
$qq="";
$add="";
if(isset($_POST['Submit']) && $_POST['Submit']=="提交"){     //对表单提交的按钮进行判断
    if(isset($_POST['user']) && $_POST['user']!=""){        //对用户名进行判断，非空时执行下面的操作
        $pass=$_POST['pass'];
        $tel=$_POST['tel'];
        $qq=$_POST['qq'];
        $add=$_POST['add'];
        $str=$_POST['user'];                                //调用用户名文本框提交的值
        $strs=ltrim($str)."\n";                             //去除用户名左边的空格
    }else{
        echo "<script>alert('提交内容不能为空');</script>";
    }
}
?>
```

DIY：去除用户填写信息时左右两边的空格和特殊符号。提示：用 trim()函数实现。（10 分）（实例位置：光盘\mr\03\qjyy\02_diy）

3.9.3　情景应用 3——对关键字进行描红

视频讲解：光盘\mr\03\lx\对关键字进行描红.exe

实例位置：光盘\mr\03\qjyy\03

对关键字描红在百度、谷歌等搜索引擎中是非常实用的，通过它可以凸显出要查找的内容。在 PHP 中完成关键字描红的操作应用 str_ireplace()函数。在本实例中通过 str_ireplace()函数完成对指定字符串的描红操作，运行结果如图 3.17 所示。

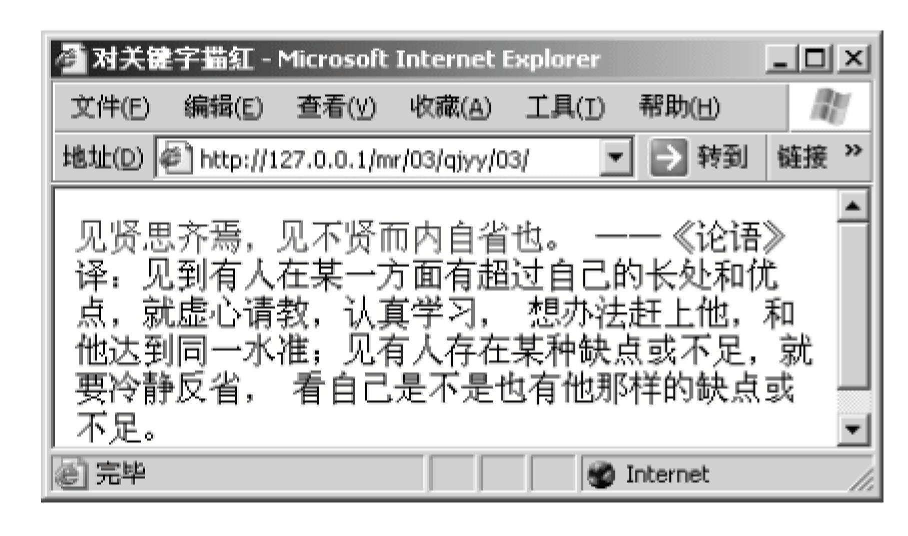


图 3.17　应用 str_ireplace()函数对查询关键字描红

str_ireplace()函数使用新的子字符串（子串）替换原始字符串中被指定要替换的字符串。该函数的语法格式如下：

```
mixed str_ireplace ( mixed search, mixed replace, mixed subject [, int &count])
```

该函数将所有在参数 subject 中出现的参数 search 以参数 replace 取代，参数&count 表示取代字符串执行的次数。

str_ireplace()函数的参数说明如表 3.8 所示。



表 3.8　str_ireplace()函数的参数说明

参　数	说　明
search	必要参数，指定需要查找的字符串
replace	必要参数，指定替换的值
subject	必要参数，指定查找的范围
count	可选参数，获取执行替换的数量

具体步骤如下：

（1）创建脚本文件并命名为 index.php。

（2）定义两个字符串变量，$content 定义查找的范围，$str 定义查询的关键字。

（3）应用 str_ireplace()对$content 中的$str 字符串进行描红操作。关键代码如下：

```
<?php
$content="见贤思齐焉，见不贤而内自省也。 ——《论语》
    译：见到有人在某一方面有超过自己的长处和优点，就虚心请教，认真学习，
想办法赶上他，和他达到同一水准；见有人存在某种缺点或不足，就要冷静反省，
看自己是不是也有他那样的缺点或不足。";
$str="见贤思齐焉，见不贤而内自省也";                                       //定义查询的字符串常量
echo str_ireplace($str,"<font color='#FF0000'>".$str."</font>",$content);         //替换字符串为红色字体
?>
```

DIY：对描红的关键字进行替换。提示：用 str_ireplace()函数实现。（10 分）（实例位置：光盘\mr\03\qjyy\03_diy）

3.9.4　情景应用 4——网页闹钟

视频讲解：光盘\mr\03\lx\网页闹钟.exe

实例位置：光盘\mr\03\qjyy\04

闹钟是生活中经常使用的一个小工具，它会在指定时间叫你起床。在 Web 程序中也可以应用这个原理，开发一个“网页闹钟”，提示用户在指定的时间或者时间段内要做什么工作。下面编写一个实例，当系统的当前时间运行到 1 月 01 日，给出提示信息“今天是元旦！”。运行结果如图 3.18 所示。



图 3.18　网页闹钟

在本实例中，通过 strtotime()函数获取系统当前时间戳，并且与 1 月 01 日的时间戳进行比较，如果相同则给出提示信息。关键代码如下：

```
<?php
$time1 = strtotime(date("Y-m-d"));                  //当前的系统时间，获取月和天的时间戳
$time2 = strtotime(date("Y")."-1-01");              //设置时间 1 月 01 日的时间戳
if($time1==$time2){                                 //判断两个时间戳是否相同
      echo "<script>alert('今天是元旦！');window.location.href='index.php';</script>";      //给出提示信息
}else{
     echo "今天不是一个特殊的日子！";
}
?>
```

说明：只有当本机的系统时间与设置的时间戳相同时，才会出现提示。

DIY：开发一个“结婚纪念日提醒”的闹钟。（10 分）（实例位置：光盘\mr\03\qjyy\04_diy）

3.9.5 情景应用 5——幸运数字抽奖

视频讲解：光盘\mr\03\lx\幸运数字抽奖.exe

实例位置：光盘\mr\03\qjyy\05

在去商场买东西时，经常会碰见商家为了扩大产品的宣传而进行的幸运抽奖小游戏，当抽到幸运的数字会给一些小的礼品作为奖励。下面应用 mt_rand()函数制作一个幸运抽奖的功能，运行结果如图 3.19 所示。



图 3.19 幸运抽奖

mt_rand()函数返回随机数中的一个值。该函数的语法格式如下：

```
int mt_rand ( [int min, int max])
```

如果 mt_rand()函数没有提供可选参数 min 和 max，则返回 0～RAND_MAX 之间的伪随机数。

创建 index.php 文件，添加 form 表单，设置一个抽奖按钮，判断当抽奖按钮的值存在，并且不为空时，应用 mt_rand()函数从 0～9 中抽取一个数字作为幸运数字输出。关键代码如下：

```
<?php
if(isset($_POST['Submit']) && $_POST['Submit']=="抽奖"){        //对表单中的按钮进行判断
        $new= mt_rand(0,9);                                      //通过 mt_rand 函数获取随机数字
    echo '幸运数字为:'.$new;                                      //输出随机数字
}
?>
```

DIY：抽取幸运号码，并输出对应的奖品。（10 分）（实例位置：光盘\mr\03\qjyy\05_diy）

3.9.6 情景应用 6——数字验证码

视频讲解：光盘\mr\03\lx\数字验证码.exe

实例位置：光盘\mr\03\qjyy\06

在用户注册过程中为了提高站点的安全性，避免由于网速慢造成用户注册信息的重复提交，往往会在用户注册表单中增加验证码功能。在本实例中，应用 mt_rand()函数从 1000～9999 的 4 位数字中随机获取用户登录时需要提交的验证码，运行结果如图 3.20 所示。



图 3.20 应用 mt_rand()函数获取随机验证码

关键代码如下：

```
<?php
$num=intval(mt_rand(1000,9999));   //应用 mt_rand()函数获取随机数
```



```
echo $num;                                //输出获取的随机数
?>
```

DIY：验证码以图片的形式出现。提示：操作过程可以参见情景应用 9 的实现过程。（10 分）（实例位置：光盘\mr\03\qjyy\06_diy）

3.9.7　情景应用 7——文本文件统计网站访问量

视频讲解：光盘\mr\03\lx\文本文件统计网站访问量.exe

实例位置：光盘\mr\03\qjyy\07

通过文本文件统计网站访问量的方法也是非常实用的一种技术，读者在以后的开发程序过程中很可能会用到。本实例联合应用文件系统函数和 GD2 函数，完成网站访问量的存储、读取和输出。运行结果如图 3.21 所示。



图 3.21　通过文本文件统计网站访问量

通过文本文件统计网站访问量，概括起来应用了 3 个方面的技术：

（1）文件系统操作函数，实现数据的写入、读取功能。

（2）SESSION，通过 SESSION 变量防止重复计数。

（3）GD2 函数，将访问量数据生成图像，改变数据的输出方式。

其实现的原理如图 3.22 所示。

下面看一下这 3 个方面的技术在程序中是如何体现的。

（1）创建 index.php 文件。通过文件操作函数和 SESSION 全局变量完成网站访问量的统计和防止重复计数，统计数据存储在实例根目录下的 counter.txt 中。通过 img 图像标签的 src 属性指定 gd1.php 文件，以图像的形式输出网站访问量的数据。关键代码如下：

```
<?php session_start();
if(!isset($_SESSION['temp'])){          //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义的变量
    if(($fp=fopen("counter.txt","r"))==FALSE){
        echo "打开文件失败!";
```

```
    }else{
        $counter=fgets($fp,1024);      //读取文件中数据
        fclose($fp);                   //关闭文本文件
        $counter++;                    //计数器增加 1
        $fp=fopen("counter.txt","w");  //以写的方式打开文本文件<!---->
        fputs($fp,$counter);           //将新的统计数据增加 1
        fclose($fp);
    }                                  //关闭文件
    $_SESSION['temp']=1;    //登录以后，$_SESSION[temp]的值不为空，给$_SESSION[temp]赋一个值 1
}
?>
<tr>
    <td height="40" align="center"><img src="gd1.php" /></td>
</tr>
```

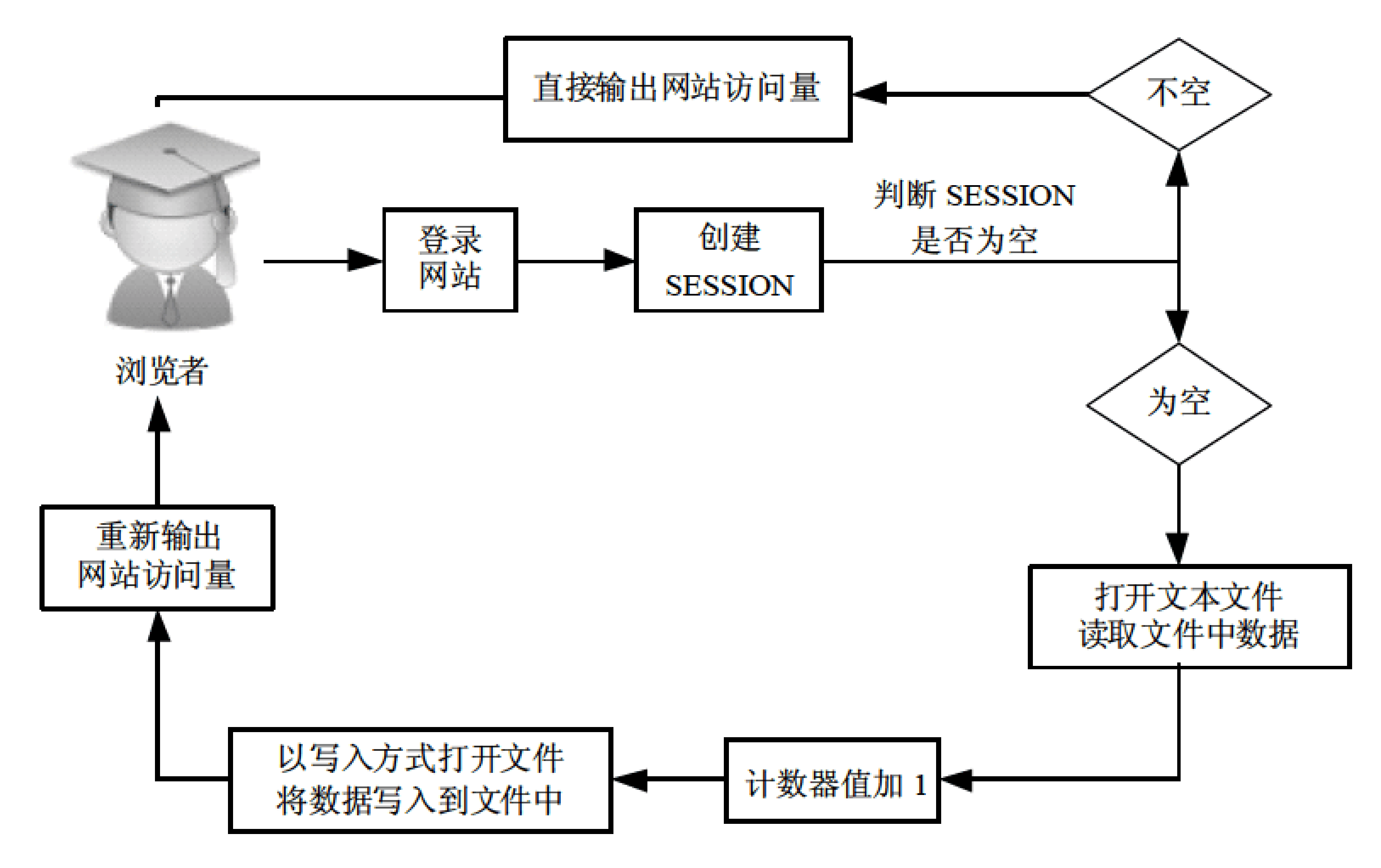


图 3.22　通过文本文件统计网站访问量设计原理

（2）创建 gd1.php 文件，根据从文本文件 counter.txt 中读取到的数据，通过 GD2 函数生成数据图像。代码如下：

```
<?php
if(($fp=fopen("counter.txt","r"))==FALSE){                    //打开文件
    echo "打开文件失败!";
}else{
    $counter=fgets($fp,1024);                                 //读取文件中数据
    fclose($fp);                                              //关闭文件
    $im=imagecreate(240,24);                                  //通过 GD2 函数创建画布
    $gray=imagecolorallocate($im,255,255,255);                //定义背景颜色
    $color =imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255));    //定义字体颜色
    //输出中文字符
    $text=iconv("gb2312","utf-8","网站的访问量:");             //对指定的中文字符串进行转换
    $font = "Fonts/FZHCJW.TTF";                               //定义使用字体的存储位置
    imagettftext($im,14,0,20,18,$color,$font,$text);          //输出中文
    imagestring($im,5,160,5,$counter,$color);                 //输出图像，输出网站的访问次数
    imagepng($im);                                            //生成 png 图像
```

```
    imagedestroy($im);                                          //销毁图像
}
?>
```

（3）在本实例的根目录下创建 counter.txt 文本文件，设置默认值为 0。同样在根目录下创建 fonts 文件夹，将使用的字体 FZHCJW.TTF 存储在该文件夹下。

DIY：以英文字符串的形式输出网站的访问量。提示：系统默认输出的是英文。（10 分）（实例位置：光盘\mr\03\qjyy\07_diy）

3.9.8　情景应用 8——购物车中商品的输出

视频讲解：光盘\mr\03\lx\购物车中商品的输出.exe

实例位置：光盘\mr\03\qjyy\08

分割字符串将指定字符串中的内容按照某个规则进行分类存储，进而实现更多的功能。例如，在电子商务网站的购物车中，可以通过特殊标识符“@”将购买的多种商品组合成一个字符串存储在数据表中，在显示购物车中的商品时，以“@”作为分割的标识符进行拆分，将商品字符串分割成 N 个数组元素，最后通过 for 循环语句输出数组元素，即输出购买的商品。本实例应用 explode()函数实现对已知的字符串“品牌电脑@品牌手机@高档男士衬衫@高档女士挎包”进行分割，运行结果如图 3.23 所示。

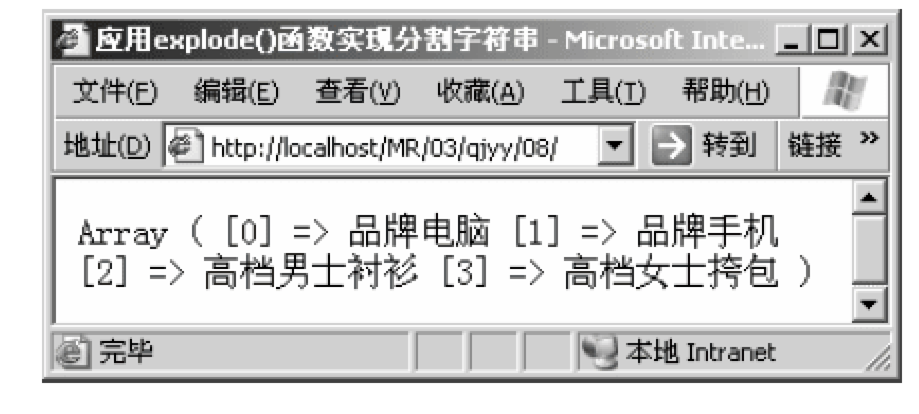


图 3.23　应用 explode()函数实现分割字符串

通过 explode()函数分割字符串的代码如下：

```
<?php
$str="品牌电脑@品牌手机@高档男士衬衫@高档女士挎包";          //定义字符串常量
$str_arr=explode("@",$str);                                //应用标识“@”分割字符串
print_r($str_arr);                                         //输出字符串分割后的结果
?>
```

从上面的代码中可以看出，在分割字符$str 时，以“@”作为分割的标识符进行拆分，分割成 4 个数组元素，最后应用 print_r()输出函数输出数组中的元素。

DIY：将数组中的元素用特定的符号连接成一个字符串输出。提示：应用 implode()函数。（10 分）（实例位置：光盘\mr\03\qjyy\08_diy）

3.9.9　情景应用 9——中文图像验证码

视频讲解：光盘\mr\03\lx\中文图像验证码.exe

实例位置：光盘\mr\03\qjyy\09

用户注册主要为了保存用户的一些基本信息，包括用户名、真实姓名、密码等相关信息，以及跟踪和定位不同用户，从而更好地服务用户。同时为了提高站点的安全性，避免由于网速慢造成用户注册信息的重复提交，往往会在用户注册表单中增加验证码功能。验证码的形式可以是数字，也可以是一些中文图像。在本实例中将汉字作为随机数的种子，应用 rand()函数从文字中随机获取值来作为验证码，运行结果如图 3.24 所示。



图 3.24 通过 rand()函数来获取文字验证码

具体步骤如下：

（1）制作中文汉字图标，以 gif 格式存储于实例根目录的 images 文件夹下。

（2）创建一个聊天室登录的 form 表单，包括用户名、密码、验证码和登录按钮。

（3）定义一个中文字符的数组，并用 count()函数计算数组中的元素数目，用 for 循环输出由 rand()函数随机获取的值，显示中文图像验证码。关键代码如下：

```
<?php
$str=array("大","更","创","天","科","客","博","技","立","新");
$word=count($str);
$img="";
$pic="";
for($i=0;$i<4;$i++){
    $num=rand(0,$word-1);                                              //生成一个 10 以内的随机数
    $img=$img."<img src='images/".$num.".gif' width='16' height='16'>";    //显示随机图片
    $pic=$pic.$str[$num];                                              //将图片转换成数组中的文字
}
?>
```

DIY：制作一个中文验证码。提示：不需要把中文字符做成图片形式，而是直接以汉字形式输出。（20 分）（实例位置：光盘\mr\03\qjyy\09_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	6	7	8	9	总分数	
分数											

3.10 自 我 测 试

一、选择题（每题 10 分，5 道题）

1．下列（　　）函数是将数组转换为字符串。

A．imploed()　　B．explode()　　C．arsort()　　D．natsort()

2．下列（　　）选项是正确地引用文件的方法。

A．require 和 include　　B．require 和 function

C．define 和 include　　D．function 和 include

3．file()函数返回的数据类型为（　　）。

A．数组　　B．字符串　　C．整型　　D．根据文件来定

4．下列（　　）函数可以打开或创建一个文件。

A．open()　　B．fopen()　　C．fwrite()　　D．write()

5．以下字符的长度是（　　）。

```
<?php
    $text="   \tllo   ";
    echo strlen(trim($text));
?>
```

A．9　　B．5　　C．7　　D．3

二、填空题（每题 10 分，5 道题）

1．输出字符串可以应用函数（　　）和（　　）来输出。

2．PHP 中用来删除当前目录的函数是（　　）。

3．哪个函数不能返回 UNIX 时间戳（　　）。

4．函数 vat_dump 的意义是（　　）。

5．下面一段代码的运行结果为（　　）。

```
<?php
    $a="php china php mrkj";
    Echo strrchr($a, "1")."<br>";
?>
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

3.11　行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强！
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	

续表

序号	内　容	行 动 指 南
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）PHP 的输出语句有 echo、print、printf、print_r。尝试使用这 4 个语句输出数据，看看它们有什么不同。
		（2）获取当前访问者的电脑信息，如 IP、端口号等。
		（3）使用 PHP 函数判断输入（这里先假定一个变量）数据是否符合下列要求：输入必须为全数字，输入字符串长度不允许超过 25，输入不允许为空。
		（4）获取指定任意一天的时间，格式为 YYYY-MM-DD HH:MM:SS。
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

3.12　成功可以复制——缔造华人的硅谷传奇杨致远

成就：创立 Yahoo!，发明了最早的网站搜索软件，“世纪网络第一人”，开启了人类的网络时代。

创始公司：Yahoo！公司，最受欢迎、效率最高的国际索引公司，资产高达 390 亿美元。

财富巅峰：1997 年荣登《财富》杂志富翁排行榜，1998 年 4 月，身价高达 80 亿美金。

最高学历：博士

超级语录：如果只是为了成功和金钱去创业，能接受失败吗？不能。怎样才能接受失败？是因为能坚持，对所做事情的热爱，一种固执的“笨”。在创业中，过程始终比终点更为重要。

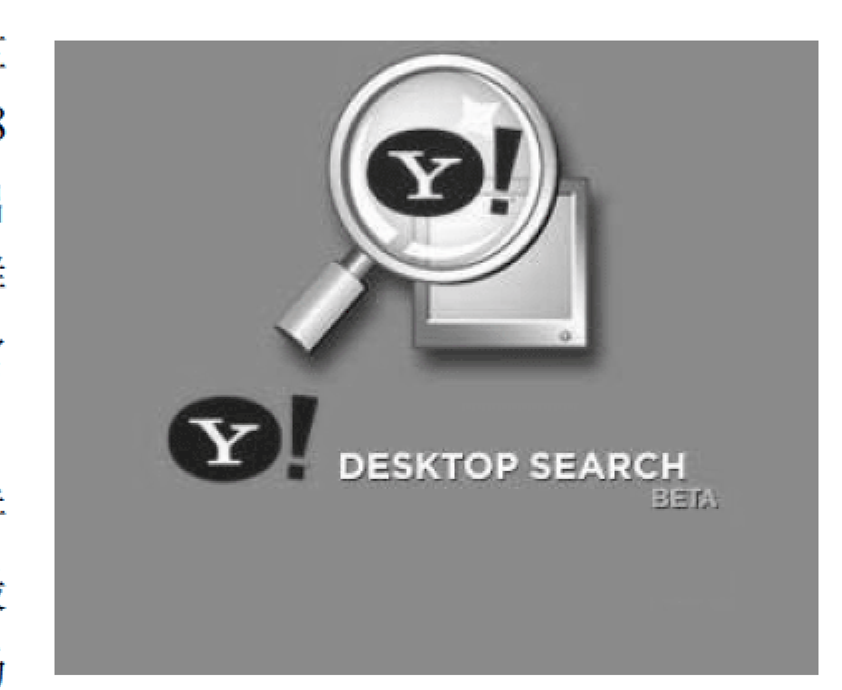
成功历程

1994 年 4 月，就读在斯坦福大学的杨致远与费洛为了完成论文，在网络上寻找资料。在这个过程中，他们也收集很多自己感兴趣的站点并加入到书签，以便查找。可是当这些所收集的站点越来越多时，管理和查找就成了大麻烦。于是他们把这些书签按类别整理好，编制成软件《Jerry’s Guide to the World Wide Web》（杰瑞全球咨讯网指南），放到网络上让其他冲浪的人享用，结果大受欢迎与拥护。

随着《指南》越来越受欢迎与关注，许多网友纷纷进入斯坦福大学电机系的工作站使用这套软件，使校方大感困扰，抱怨这项发明影响了学校电脑的正常运作。但是杨致远与费洛仍然积极为此努力，并推出特色栏目：Cool Links 和 Hard to Believe，并将网站改名为《Jerry and David’s Guide to the World Wide Web》。由于但是条件限制，网站的数据和搜索引擎分别放在他们个人的计算机中。

在没日没夜、满目狼藉的工作室中，1994 年夏天的一个晚上，杨致远与费洛为他们的网站取了一个很特殊的名字：Yahoo!，于是神奇的 Yahoo! 网站诞生了，当时是凌晨 2 点。由于 Yahoo! 检索系统确实方便，前景被普遍看好，广告收入也相当乐观。结果，Yahoo! 一上市就一鸣惊人，风头大出。

1996 年 4 月 12 日，Yahoo! 股票首次上市，每股价为$13，而开盘就达到$24.50，并持续飙升至$43。一下子给 Yahoo! 公司带来将近 8.48 亿美元的市场值。创造了纳斯达克纪录。1998 年《福布斯》杂志推出高科技百名富翁，杨致远以 10 亿美元的财富跃居第 16 位，超过了冠群 CEO 王嘉廉，成为高科技中的华人首富。1999 年，杨致远的纸面财富更是达到 75 亿美元。



Yahoo!的成功，与杨致远清晰快捷的思维，不知疲倦地为事业打拼是分不开的。在当今 IT 业，有 3 位没有上完大学就跑出去开公司、最后变成著名的亿万富翁的人，他们就是比尔・盖茨、迈克・戴尔和杨致远。

深度评价

“我不怕输，即使我失败，我也有重新来过的勇气。”杨致远这种不怕输的劲头也衬托出一个传奇人物的精髓所在。引用杨致远的一句话：机会是偶然的，但取得成功并非如此，你把一种嗜好变成了一个事业，这需要巨大的努力。

第 4 堂课

PHP 流程控制语句

（视频讲解：95 分钟）

条件控制语句和循环控制语句是两种基本的语法结构，它们都是用来控制程序执行流程的，也是构成程序的主要语法基础。通常，语句是按顺序执行的，即执行完当前的语句之后，执行下一条语句。但是这种简单的顺序执行方式能够解决的问题很有限，现实生活中的许多逻辑使用顺序执行的方式是不能够被描述的。例如，“如果明天不下雨，我们就去放风筝，否则我们呆在家里练习书法”。在程序中需要根据条件判断来确定是“放风筝”还是“呆在家里练习书法”，两种结果不可能同时存在，这里只能应用条件控制语句实现这个功能，而循环控制语句不能够确定选择哪种结果。因此，开发人员可以按照实际问题的逻辑思路选择性地使用流程控制语句来编写程序代码。

学习摘要：

- **程序的 3 种控制结构**
- **条件控制语句**
- **循环控制语句**
- **跳转语句**
- **包含语句**

4.1　程序的 3 种控制结构

在编程的过程中，所有的操作都是在按照某种结构有条不紊地进行，学习 PHP 语言，不仅要掌握其中的函数、数组和字符串等实际的知识，更重要的是通过这些知识形成一种属于自己的编程思想和编程方法。要想形成属于自己的编程思想和方法，那么首先就要掌握程序设计的结构，再配合以函数、数组和字符串等实际的知识，逐步形成一种属于自己的编程方法。

程序设计的结构大致可以分为顺序结构、选择结构和循环结构 3 种。在对这 3 种结构的使用中，几乎很少有哪个程序是单独地使用某一种结构来完成某个操作，基本上都是其中的两种或者 3 种结构结合使用。为了更好地应用这 3 种程序控制结构，下面进行详细介绍。

4.1.1　顺序结构

顺序结构是最简单基本的结构方式，各流程框依次按顺序执行。传统流程图的表示方式与 N-S 结构化流程图的表示方式分别如图 4.1 和图 4.2 所示。执行顺序为：开始→语句 1→语句 2→…→结束。

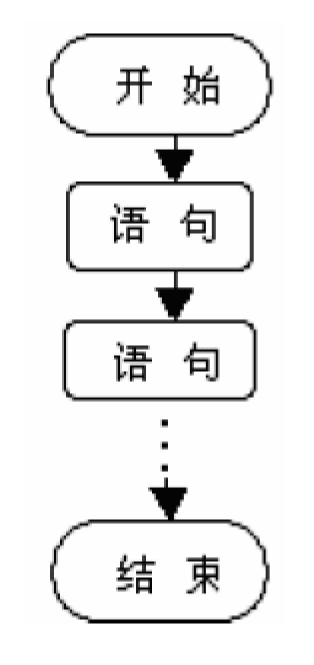


图 4.1　顺序结构传统流程图



图 4.2　N-S 结构化流程图

4.1.2　选择（分支）结构

选择结构就是对给定条件进行判断，条件为真时执行一个分支，条件为假时执行另一个分支。其传统流程图表示方式与 N-S 结构化流程图表示方式分别如图 4.3 和图 4.4 所示。

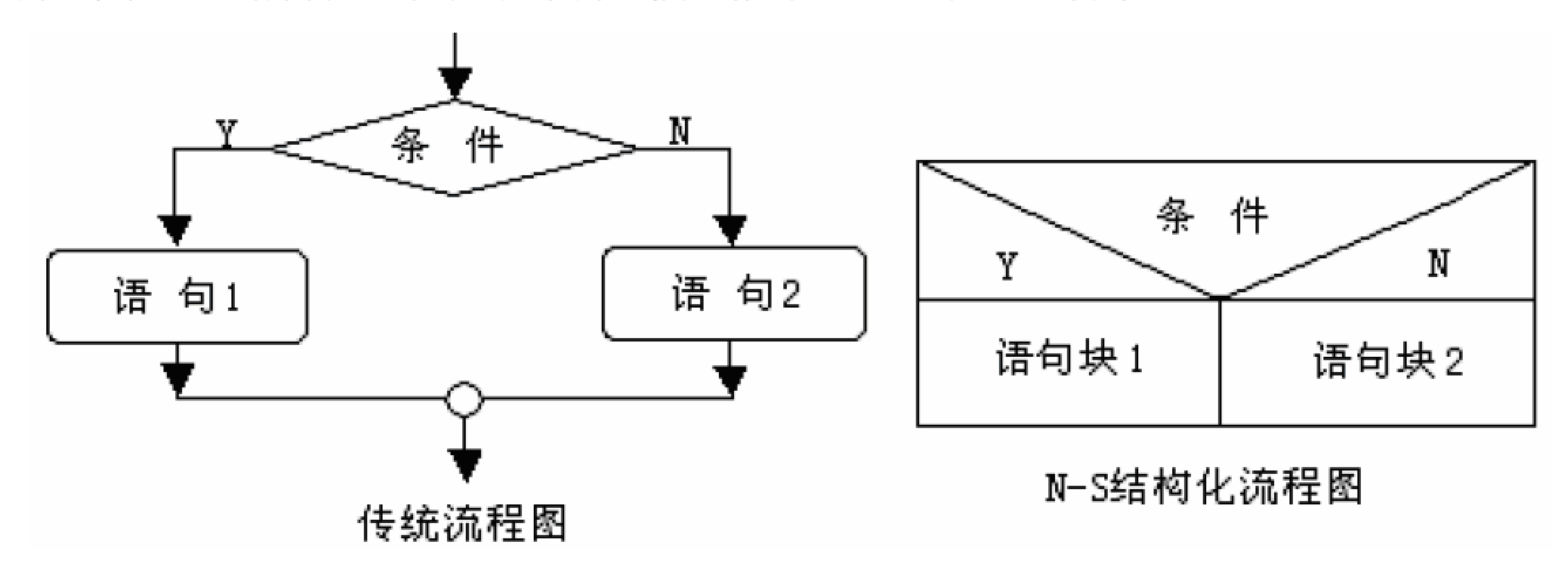


图 4.3　条件成立与否都执行语句或语句块

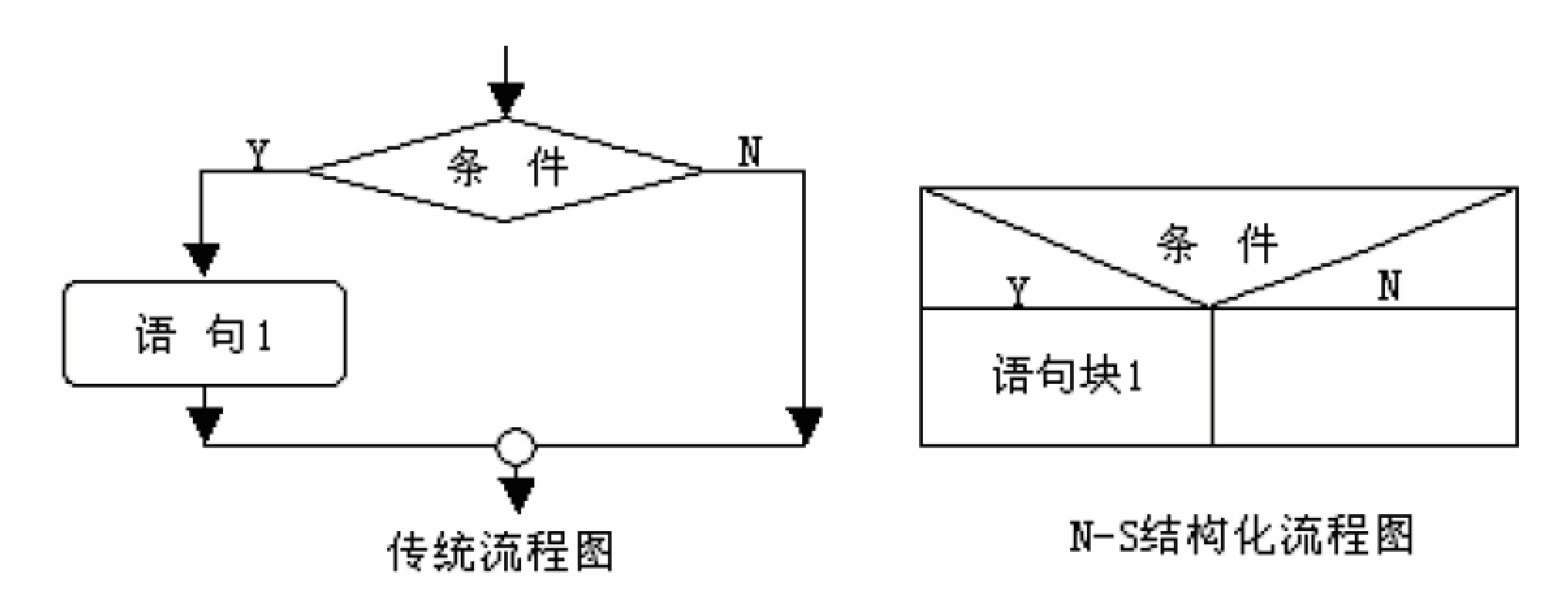


图 4.4　条件为否不执行语句或语句块

4.1.3　循环结构

循环结构可以按照需要多次重复执行一行或者多行代码。循环结构分为前测试型循环和后测试型循环两种。

前测试型循环，先判断后执行。当条件为真时反复执行语句或语句块；条件为假时，跳出循环，继续执行循环后面的语句，流程图如图 4.5 所示。

后测试型循环，先执行后判断。先执行语句或语句块，再进行条件判断，直到条件为假时，跳出循环，继续执行循环后面的语句，否则一直执行语句或语句块，流程图如图 4.6 所示。

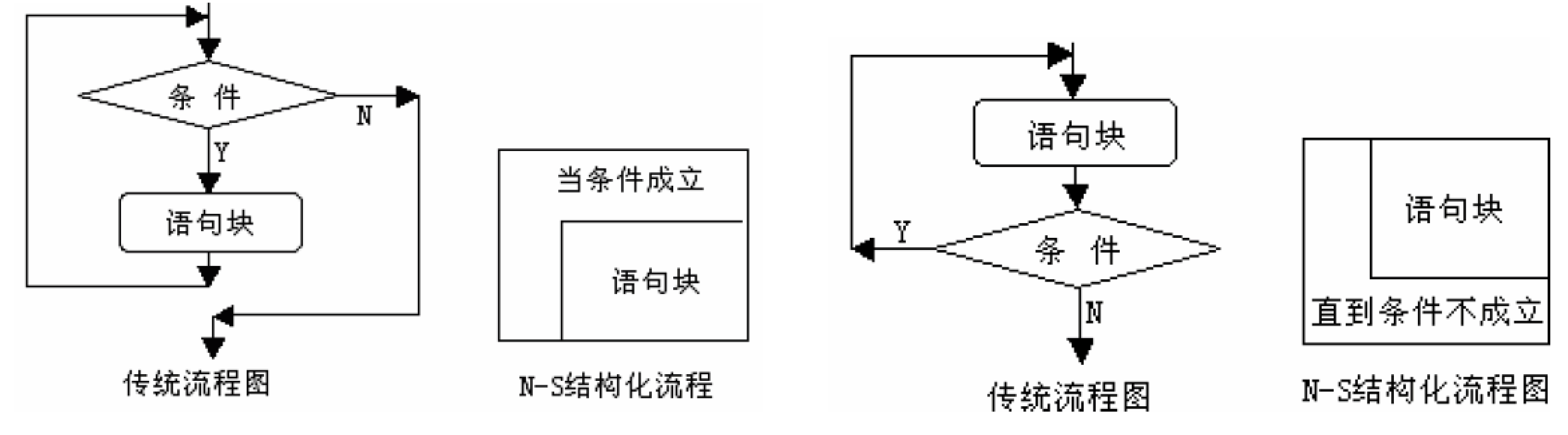


图 4.5　前测试型循环流程图　　　　图 4.6　后测试型循环流程图

在 PHP 中，大多数情况下程序都是以这 3 种结构的组合形式出现。其中，顺序结构很容易理解，就是直接输出程序运行结果，而选择和循环结构则需要一些特殊的控制语句来实现，包括以下 3 种控制语句。

- ☑　条件控制语句：if、else、elseif 和 switch。
- ☑　循环控制语句：while、do…while、for 和 foreach。
- ☑　跳转控制语句：break、continue 和 return。

下面将对这 3 种控制语句进行详细的讲解。

4.2　条件控制语句

所谓条件控制语句就是对语句中不同条件的值进行判断，进而根据不同的条件执行不同的语句。在条件控制语句中主要有两个语句：if 条件控制语句和 switch 多分支语句。

4.2.1　if 条件控制语句

if 条件控制语句是所有流程控制语句中最简单、最常用的一个，根据获取的不同条件判断执行不同的语句。应用范围十分广泛，无论程序大小几乎都会应用到该语句。其语法如下：

```
if (expr)
    statement ;              //这是基本的表达式
if () {}                     //这是执行多条语句的表达式
if () {}else {}              //这是通过 else 延伸了的表达式
if () {}elseif() {} else {}  //这是加入了 elseif 同时判断多个条件的表达式
```

参数 expr 按照布尔求值。如果 expr 的值为 TRUE，将执行 statement；如果值为 FALSE，则忽略 statement。if 语句可以无限层地嵌套到其他 if 语句中去，实现更多条件的执行。

else 的功能是当 if 语句在参数 expr 的值为 FALSE 时执行其他语句，即在执行的语句不满足该条件时执行 else 后大括号中的语句。

在同时判断多个条件时，PHP 提供了 elseif 的语句来扩展需求。elseif 语句被放置在 if 和 else 语句之间，满足多条件同时判断的需求。

if 语句的流程如图 4.7～图 4.9 所示。

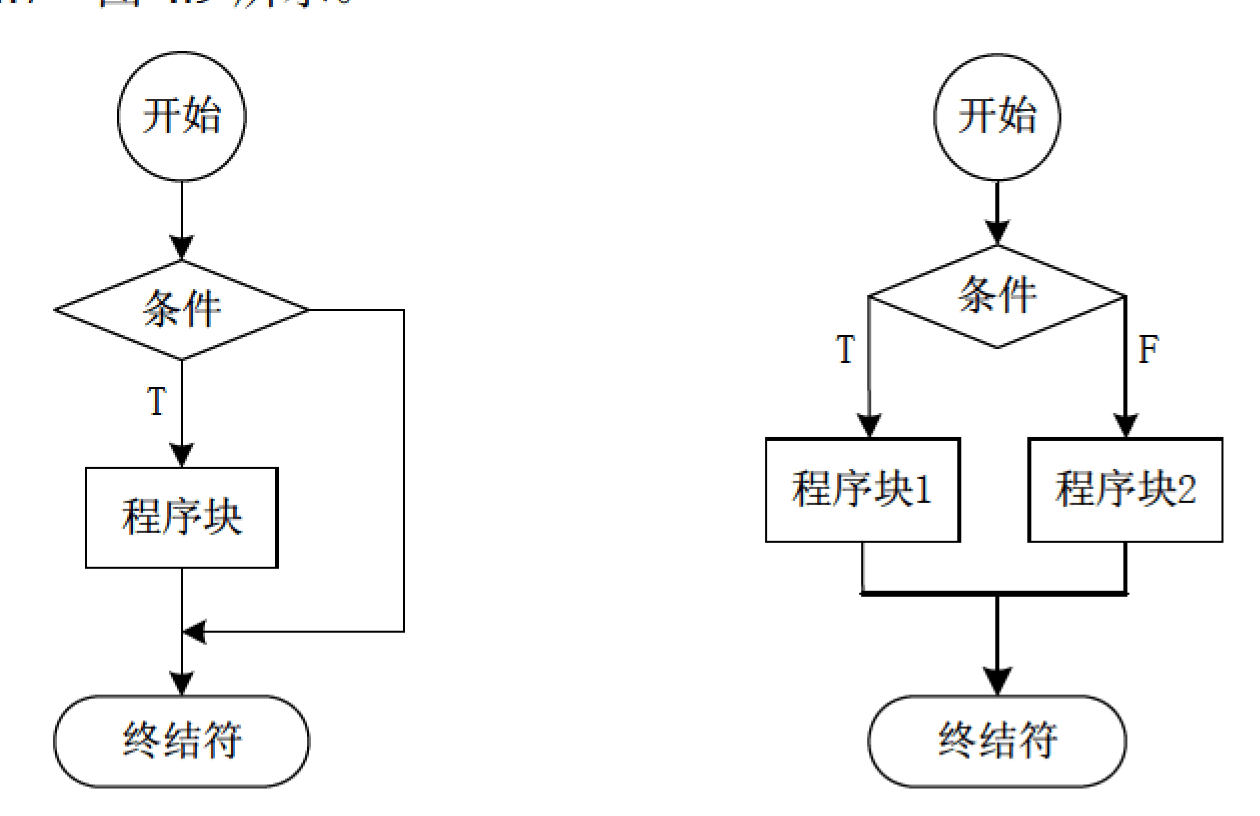


图 4.7　if 语句流程图　　图 4.8　if…else 语句流程控制图

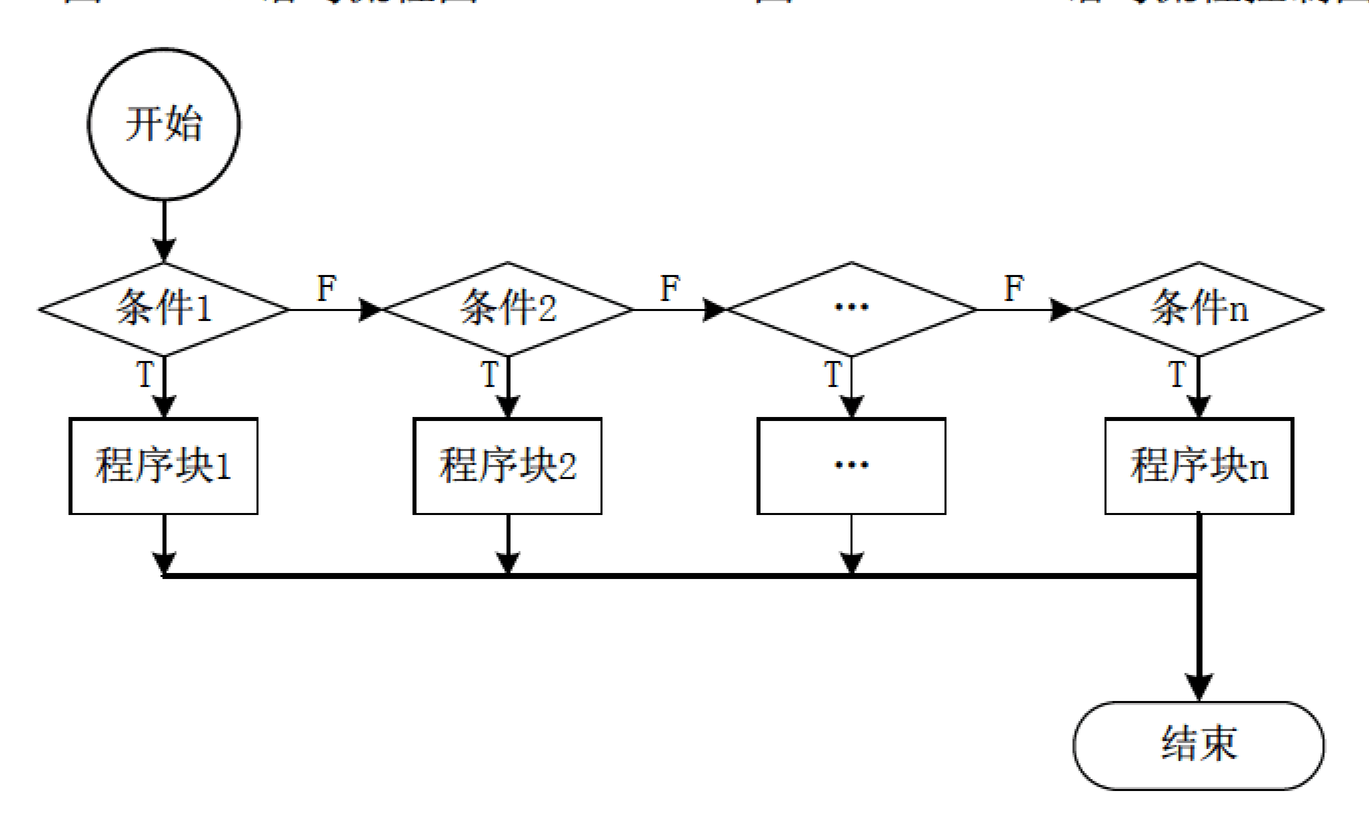


图 4.9　elseif 语句的流程控制图

下面以实例的形式展示 if 条件语句强大的功能。



例 4.01　通过 if 语句判断用户提交的登录信息是否为空。（**实例位置：光盘\mr\04\sl\4.01**）

具体步骤如下：

创建 index.php 文件，创建一个用户登录页面，提交登录的用户名和密码。然后在页面中通过$_POST[]方法获取表单中提交的用户名和密码，并且应用 if 语句判断用户提交的登录信息是否为空。关键代码如下：

```
        <?php
/*
$_POST[]方法获取表单提交的按钮“sub”、用户名“textt”和密码“pwd”的值
Isset()函数检测按钮变量 sub 是否存在。如果存在返回 TRUE，否则返回 FALSE
$_POST[text]!=""||$_POST[pwd]!=""判断用户名和密码是否为空，不为空返回 TRUE，否则返回 FALSE
测试成功通过 echo 语句返回提示信息
*/
if(isset($_POST['sub'])){
        if($_POST['text']!=""||$_POST[pwd]!=""){
                echo "<script>alert('测试成功');</script>";
        }else{
                echo "<script>alert('文本框内容不能为空');</script>";
        }
}
?>
```

运行结果如图 4.10 所示。



图 4.10　用户登录模块的实现

4.2.2　switch 多分支语句

switch 语句和 if 条件控制语句类似，实现将同一个表达式与很多不同的值比较，获取相同的值，并且执行相同的值对应的语句。其语法如下：

```
<?php
switch ( expr ){                        //expr 条件为变量名称
            case expr1:                 //case 后的 expr1 为变量的值
                  statement1;           //冒号":"后的是符合该条件时要执行的部分
              break ;                   //应用 break 来跳离循环体
            case expr2 :
                  statement2 ;
              break ;
          default:
              statementN;
```

```
            break;
}
?>
```

参数说明如表 4.1 所示。

表 4.1　switch 语句参数说明

参　　数	说　　明
expr	表达式的值，即 switch 语句的条件变量的名称
expr1	放置于 case 语句之后，是要与条件变量 expr 进行匹配的值中的一个
statement1	在参数 expr1 的值与条件变量 expr 的值相匹配时执行的代码
break 语句	终止语句的执行，即当语句在执行过程中，遇到 break 就停止执行，跳出循环体
default	case 的一个特例，匹配任何其他 case 都不匹配的情况，并且是最后一条 case 语句

switch 语句的流程控制如图 4.11 所示。

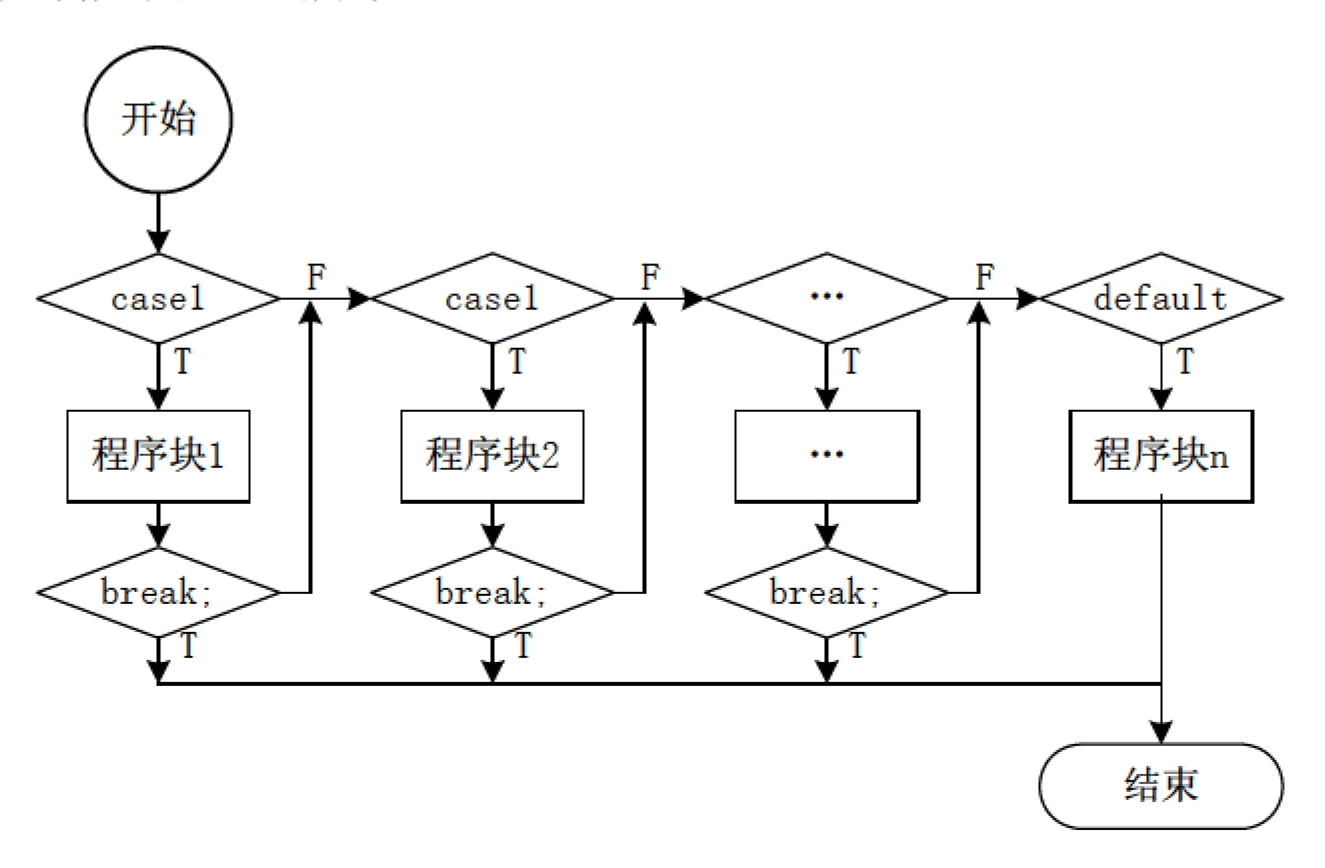


图 4.11　switch 语句流程控制图

上述讲解的是 switch 语句的工作流程，下面编写一个小实例，看一下 switch 语句在实际中的应用。

例 4.02　应用 switch 语句判断成绩的等级情况，代码如下。（**实例位置：光盘\mr\04\sl\4.02**）

```
<?php
    $cont=49;                    //以下代码实现了根据$cont 的值，判断成绩等级的功能
    switch($cont) {
        case $cont==100;         //如果$cont 的值等于 100，则输出“满分”
            echo"满分";
            break;
        case $cont>=90;          //如果$cont 的值大于等于 90，则输出“优秀”
            echo"优秀";
            break;
        case $cont>=60;          //如果$cont 的值大于等于 60，则输出“及格”
            echo"及格";
            break;
        default:                 //如果$cont 的值小于 60，则输出“不及格”
            echo"不及格";
    }
?>
```

你问我答：if 语句与 switch 语句的区别？

if 和 switch 语句可以从使用的效率上来进行区别，也可以从实用性角度去区分。如果从使用的效率上进行区分，在对同一个变量的不同值做条件判断时，使用 switch 语句的效率相对更高一些，尤其是判断的分支越多越明显。

如果从语句的实用性的角度去区分，那 switch 语句肯定不如 if 条件语句。if 条件语句是实用性最强和应用范围最广的语句。

在程序开发的过程中，if 和 switch 语句的使用应该根据实际情况而定，不要因为 switch 语句的效率高就一味地使用，也不要因为 if 语句常用就不应用 switch 语句。要根据实际情况，具体问题具体分析，使用最适合的条件语句。一般情况下可以使用 if 条件语句，但是在实现一些多条件的判断中，特别是在实现框架的功能时就应该使用 switch 语句。

4.3　循环控制语句

循环语句是在满足条件的情况下反复地执行某一个操作。在 PHP 中，提供了 4 个循环控制语句，分别是 while 循环语句、do…while 循环语句、for 循环语句和 foreach 循环语句。

4.3.1　while 循环语句

while 循环语句，其作用是反复地执行某一项操作，是循环控制语句中最简单的一个，也是最常用的一个。while 循环语句对表达式的值进行判断，当表达式为非 0 值时，执行 while 语句中的内嵌语句；当表达式的值为 0 值时，则不执行 while 语句中的内嵌语句。该语句的特点是：先判断表达式，后执行语句。while 循环控制语句的操作流程如图 4.12 所示。

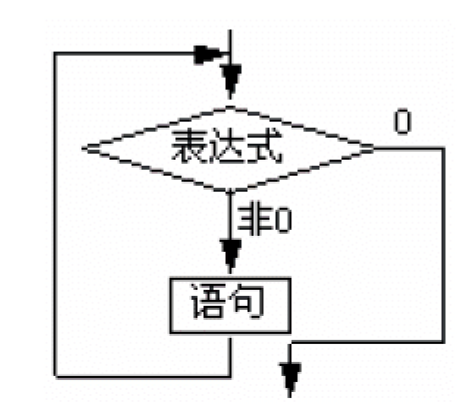


图 4.12　while 循环控制语句的操作流程

其语法如下：

```
while (expr){                              /*
        statement;                         先判断条件，当条件满足时执行语句块，否则不向下执行
}                                          */
```

只要 while 表达式 expr 的值为 TRUE，就重复执行嵌套中的 statement 语句，如果 while 表达式的值一开始就是 FALSE，则循环语句一次也不执行。

例 4.03　下面展示 while 语句的应用，计算员工的工龄工资。正式员工在单位工作每增加一年，工龄工资增长 40，以 10 年为上限，计算一个员工总的工龄工资增加情况，核心代码如下。（**实例位置：光盘\mr\04\sl\4.03**）

```
<?php
$a=1;
$year=10;
while($a<=$year){
$price=40*12*$a;
        echo "您第".$a."年的工龄工资为<b>".$price."</b>元<br>";
        $a++;
}
?>
```

运行结果如图 4.13 所示。

计算员工的工龄工资

您第1年的工龄工资为600元
您第2年的工龄工资为1200元
您第3年的工龄工资为1800元
您第4年的工龄工资为2400元
您第5年的工龄工资为3000元
您第6年的工龄工资为3600元
您第7年的工龄工资为4200元
您第8年的工龄工资为4800元
您第9年的工龄工资为5400元
您第10年的工龄工资为6000元

图 4.13　计算员工的工龄工资

4.3.2　do…while 循环语句

do…while 语句也是循环控制语句中的一种，使用方式和 while 相似，也是通过判断表达式的值来输出循环语句。其语法如下：

```
do{                              /*
                     程序在未经判断之前就进行了一次循环，循环到 while 部分才判断
               statement;                  条件，即使条件不满足，程序也已经运行了一次
*/
}while(expr);
```

该语句的操作流程是：先执行一次指定的循环体语句，然后判断表达式的值，当表达式的值为非 0 时，返回重新执行循环体语句，如此反复，直到表达式的值等于 0 为止，此时循环结束。其特点是先执行循环体，然后判断循环条件是否成立。do…while 循环语句的操作流程如图 4.14 所示。

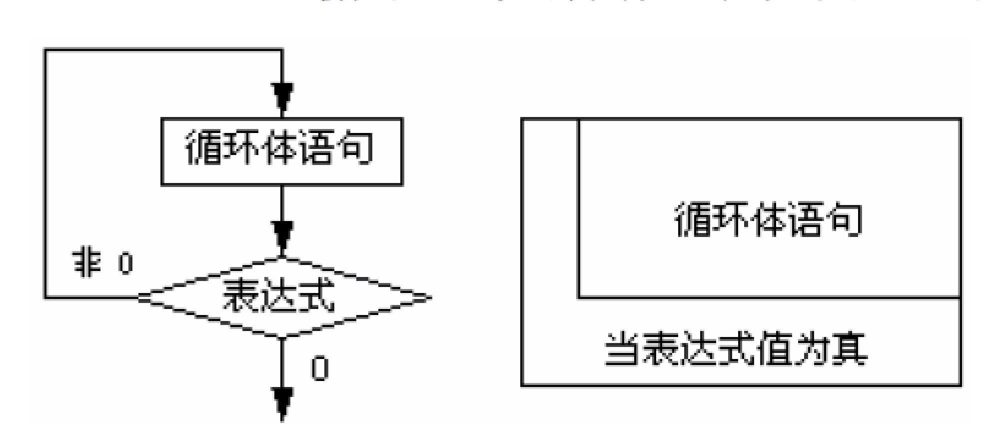


图 4.14　do…while 循环语句的操作流程

你问我答：while 和 do…while 语句之间有什么区别？

do…while 语句是先执行后判断，无论表达式的值是否为 TRUE，都将执行一次循环。而 while 语句则是首先判断表达式的值是否为 TRUE，如果为 TRUE 则执行循环语句；否则将不执行循环语句。

例 4.04　下面通过实例介绍 do…while 语句和 while 语句的区别，同样以计算员工的工龄工资为例，判断当员工工作到第 10 个年头，是否还有工龄工资。

如果使用 while 语句进行判断，那么当变量$year 的值为 10 时，已经超出表达式的值，将不会再继续计算工龄工资。如果使用 do…while 语句进行判断，那么当变量$year 的值为 10 时，程序没有停止，它将会继续输出员工第 11 年的工龄工资后停止，核心代码如下。（**实例位置：光盘\mr\04\sl\4.04**）



```
<?php
      $a=11;                  //当直接定义变量$a 的值为 11 时，仍可以输出第 11 年的
      $year=10;               //定义初始变量$year=10
      do{
            $price=40*12*$a;
            echo "您第".$a."年的工龄工资为<b>".$price."</b>元<br>";
            $a++;
      }while($a<=$year);  //当$year 等于 10 时程序没有停止，继续计算第 11 年工资，当$year 等于 11 时判断条
件不符合停止循环，但是第 11 年的工资已经输出了。
      ?>
```

运行结果如图 4.15 所示。

计算员工的工龄工资

您第11年的工龄工资为6600元

图 4.15　计算员工的工龄工资

说明： 编写这个实例意在说明 while 语句与 do…while 语句在执行判断上的一个小小区别，在实际的程序开发中是不会出现上述的这种情况的。

4.3.3　for 循环语句

for 语句是 PHP 中最复杂的循环控制语句，拥有 3 个条件表达式。其语法如下：

```
for (expr1; expr2; expr3){
     statement
}
```

for 循环语句的参数说明如表 4.2 所示。

表 4.2　for 循环语句的参数说明

参　数	说　明
expr1	必要参数。第一个条件表达式，在第一次循环开始时被执行
expr2	必要参数。第二个条件表达式，在每次循环开始时被执行，决定循环是否继续
expr3	必要参数。第 3 个条件表达式，在每次循环结束时被执行
statement	必要参数。满足条件后，循环执行的语句

其执行的过程：首先执行表达式 1；然后执行表达式 2，并对表达式 2 的值进行判断，如果值为真，则执行 for 循环语句中指定的内嵌语句，如果值为假，则结束循环，跳出 for 循环语句；最后执行表达式 3（切忌是在表达式 2 的值为真时），返回表达式 2 继续循环执行。for 循环语句的操作流程如图 4.16 所示。

例 4.05　下面使用 for 循环来计算 2～100 之间所有偶数之和，核心代码如下。**（实例位置：光盘\mr\04\sl\4.05）**

```
<?php
      $b="";
      for($a=0;$a<=100;$a+=2){
            $b=$a+$b;
      }
```

```
    echo "结果为：<b>".$b."</b>";
?>
```

运行结果如图 4.17 所示。

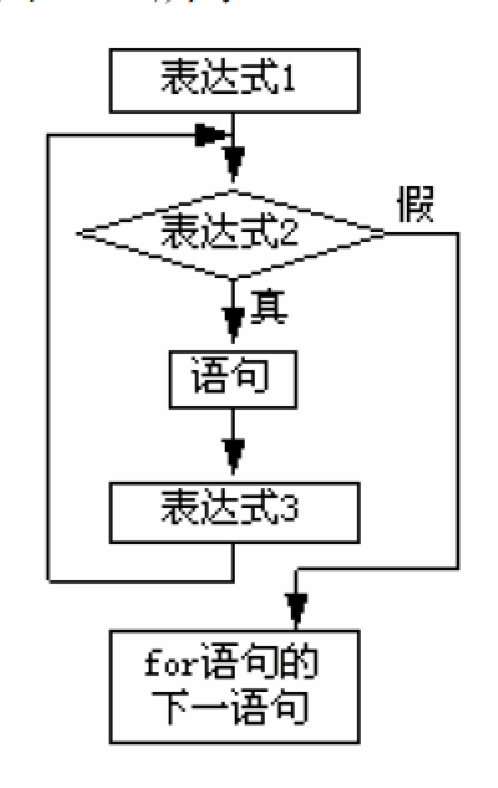


图 4.16　for 循环语句的操作流程

计算2~100所有偶数的和

结果为：2550

图 4.17　计算 2～100 之间所有偶数之和

说明： 在编程时，有时会遇到使用 for 循环的特殊语法格式来实现无限循环。其语法格式如下：

```
for(;;){
   …
}
```

对于这种无限循环可以通过 break 语句跳出循环。例如：

```
for(;;){
    if(x <20)
        break;
    x++;
}
```

4.3.4　foreach 循环语句

foreach 循环控制语句自 PHP 4 开始被引入，主要用于处理数组，是遍历数组的一种简单方法。如果将该语句用于处理其他的数据类型或者初始化的变量，将会产生错误。该语句的语法有以下两种格式：

```
foreach (array_expression as $value){
    statement
}
```

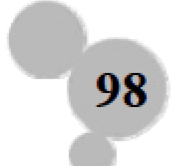
或

```
foreach (array_expression as $key => $value){
    statement
}
```

参数 array_expression 是指定要遍历的数组，其中，$value 是数组的值，$key 是数组的键名；statement 是满足条件时要循环执行的语句。

在第一种格式中，当遍历指定的 array_expression 数组，每次循环时，将当前数组单元的值赋给变量$value，并且将数组中的指针移动到下一个单元。

在第二种格式中的应用是相同的，只是在将当前单元的值赋给变量$value 的同时，将当前单元的键名也



赋给了变量$key。

> 注意：当使用 foreach 语句用于其他数据类型或者未初始化的变量时会产生错误。为了避免这个问题，最好使用 is_array()函数先来判断变量是否为数组类型，如果是，再进行其他操作。

例 4.06　下面应用 foreach 语句处理一个数组，实现输出购物车中商品的功能。这里假设将购物车中的商品存储于指定的数组中，然后通过 foreach 语句来输出购物车中的商品信息。**（实例位置：光盘\mr\04\sl\4.06）**

具体步骤如下：

（1）定义 3 个数组，用于存储商品信息。

（2）利用 foreach 语句遍历数组并循环输出数组元素。关键代码如下：

```
<?php
/*
        PHP 中的数组元素较其他编程语言有所不同，PHP 中的数组下标可以为数字，默认情况下数组下标以
0 开始，数组下标还可以使用字符串作为数组键值，至于具体的细节请参照本书的数组课程讲解。
*/
$name = array("1"=>"钢笔","2"=>"衬衫","3"=>"手机","4"=>"电脑");                //定义数组并赋值
$price = array("1"=>"108 元","2"=>"88 元","3"=>"666 元","4"=>"6666 元");
$counts = array("1"=>1,"2"=>1,"3"=>2,"4"=>1);
echo '<table width="480" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#FF0000">
<tr>
    <td width="144" align="center" bgcolor="#FFFFFF"  class="STYLE1">商品名称</td>
    <td width="144" align="center" bgcolor="#FFFFFF"  class="STYLE1">价  格</td>
    <td width="144" align="center" bgcolor="#FFFFFF"  class="STYLE1">数量</td>
    <td width="144" align="center" bgcolor="#FFFFFF"  class="STYLE1">金额</td>
</tr>';
foreach($name as $key=>$value){                    //使用 foreach 语句遍历数组，输出键和值
   echo '<tr>
      <td height="24" align="center" bgcolor="#FFFFFF">'.$value.'</td>
      <td align="center" bgcolor="#FFFFFF">'.$price[$key].'</td>
      <td align="center" bgcolor="#FFFFFF">'.$counts[$key].'</td>
      <td align="center" bgcolor="#FFFFFF">'.$counts[$key]*$price[$key].'</td>
</tr>';
}
echo '</table>';
?>
```

运行结果如图 4.18 所示。



商品名称	价 格	数量	金额
钢笔	108元	1	108
衬衫	88元	1	88
手机	666元	2	1332
电脑	6666元	1	6666

图 4.18　应用 foreach 语句输出购物车中的商品

4.4 跳 转 语 句

跳转语句主要分为 break 语句、continue 语句和 return 语句 3 个部分。其中前两个跳转语句使用起来非常简单，而且非常容易掌握，主要原因是它们都被应用在指定的环境中，如 for 循环语句中。return 语句在应用环境上较前两者相对单一，一般被用在自定义函数和面向对象的类中。

4.4.1 break 跳转语句

break 关键字可以终止当前的循环，包括 while、do…while、for、foreach 和 switch 在内的所有控制语句。

break 语句不仅可以跳出当前的循环，还可以指定跳出几重循环。其格式如下：

```
break   n;
```

参数 n 指定要跳出的循环数量。break 关键字的流程图如图 4.19 所示。

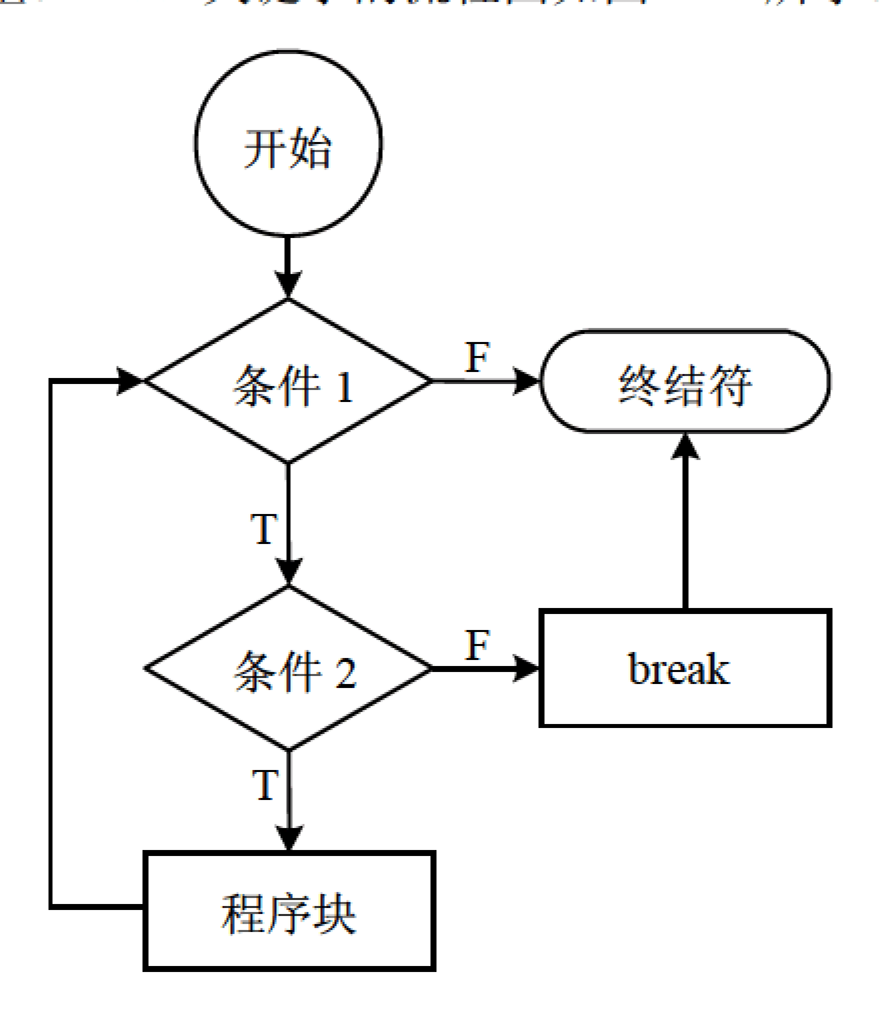


图 4.19 break 关键字的流程图

例 4.07 应用 for 循环控制语句声明变量$i，循环输出 4 个表情头像，当变量$i 等于 4 时，使用 break 跳转控制语句跳出 for 循环，代码如下。**（实例位置：光盘\mr\04\sl\4.07）**

```
<?php
for($i=1;$i<=4;$i++){                    //应用 for 循环控制语句输出表情头像
     if($i==4){                          //判断变量是否等于 4
          break;                         //如果等于 4，使用 break 语句跳出循环
     }
?>
<input type="radio" name="head" value="<?php echo("images/".$i.".jpg");?>" />
<img src="<?php echo("images/".$i.".jpg");?>" width="90" height="90" id="head"/>
<?php
   }
?>
```

运行结果如图 4.20 所示。





图 4.20　应用 break 跳转控制语句跳出循环

4.4.2　continue 跳转语句

在程序执行 break 后，程序将跳出循环，而开始继续执行循环体的后续语句。continue 跳转语句的作用没有 break 那么强大，只能终止本次循环，而进入到下一次循环中。在执行 continue 语句后，程序将结束本次循环的执行，并开始下一轮循环的执行操作。continue 也可以指定跳出几重循环。continue 跳转语句的流程图如图 4.21 所示。

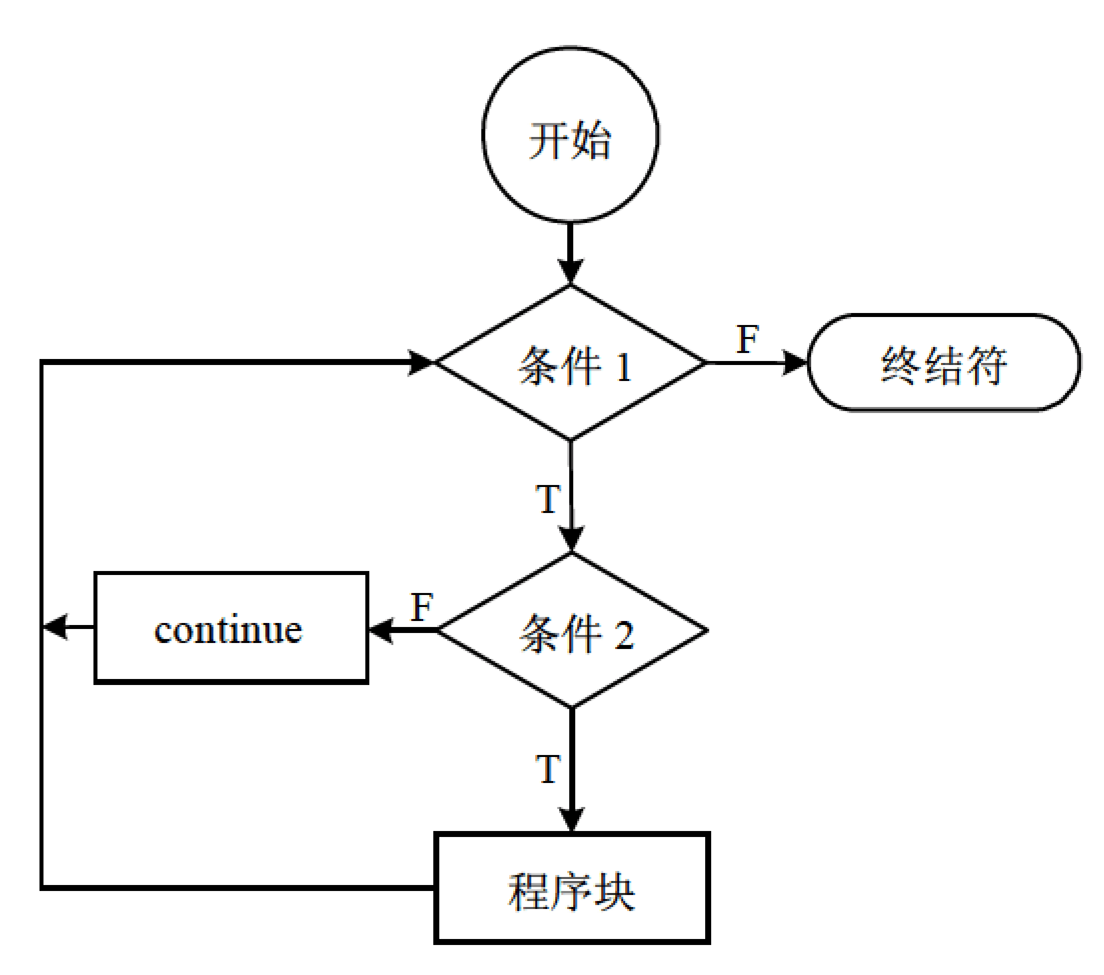


图 4.21　continue 跳转语句的流程图

例 4.08　使用 for 循环输出数组变量。如果变量的数组下标为偶数，则只输出一个空行；如果是奇数，则继续输出。在最里面的循环中，判断当前数组下标是否等于$i，如果不相等，输出数组变量，否则跳到最外重循环，代码如下。**（实例位置：光盘\mr\04\sl\4.08）**

```
<?php
$arr = array("PHP 程序开发范例宝典","PHP 典型模块开发大全","PHP 函数参考大全","PHP 项目开发全程实录",
"PHP 从入门到精通","PHP 网络编程自学手册");              //声明一个数组变量$arr
for($i = 0; $i < 6; $i++){                               //使用 for 循环
    echo "<br>";
    if($i % 2 == 0){                                      //如果$i 的值为偶数，则跳出本次循环
        continue;
    }
    for(;;){                                              //无限循环
        for($j = 0; $j < count($arr); $j++){              //再次使用 for 循环输出数组变量
            if($j == $i){                                 //如果当前输出的数组下标等于$i
```

```
                    continue 3;                              //跳出最外重循环
                }else{
                    echo $arr[$j]."\t | ";                   //输出表达式
                }
            }
        }
echo "你永远都看不到我噢!";
}
?>
```

运行结果如图 4.22 所示。

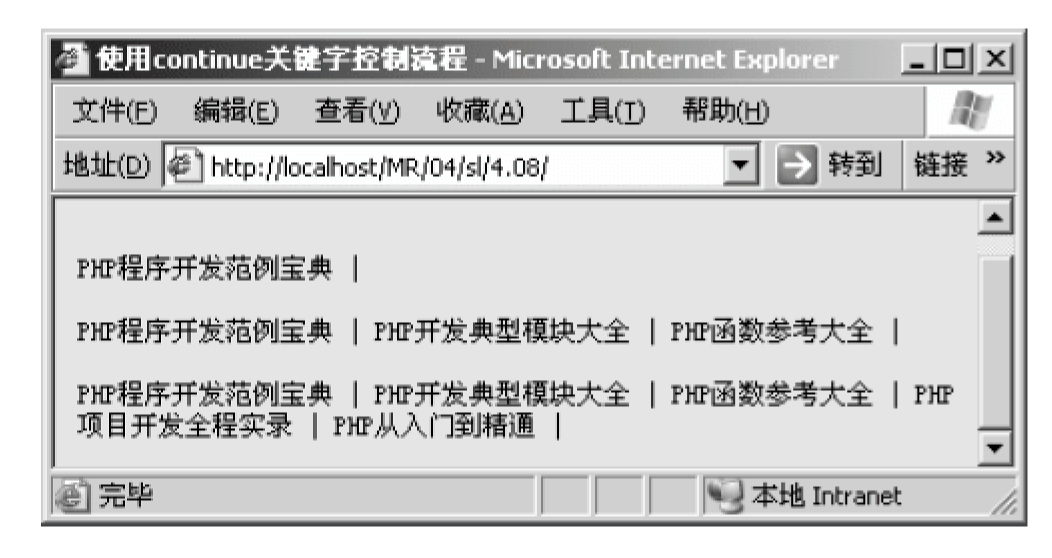


图 4.22　使用 continue 关键字控制流程

说明：同样 break 和 continue 语句都是实现跳转的功能，但还是有区别：continue 语句只是结束本次循环，并不是终止整个循环的执行。而 break 语句则是结束整个循环过程，不再判断执行循环的条件是否成立。它们之间的区别可以参考这两个语句的操作流程图。

你问我答：什么是流程控制语句的嵌套？

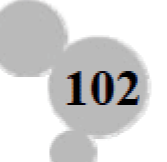
嵌套语句是在流程控制语句中又嵌入控制语句，就是在大圈中套中圈，中圈中又套了一个小圈。其中嵌套的层次没有限制，如果程序需要可以无限地嵌套下去，只是过多地使用嵌套语句会影响程序运行的速度。

通过嵌套语句可以使程序更加有层次感。嵌套语句的执行顺序是从外到内。嵌套语句可以分为两种类型。一种是同一类型语句中的嵌套，即反复地使用相同的控制语句实现嵌套的功能。

另一种是不同的控制语句之间进行嵌套使用，这种方式是最常用的，几乎所有的程序都使用过这种嵌套方法。例如，在 if 条件语句中嵌套 for 循环语句、while 循环语句；在 for 循环语句中嵌套 while 循环语句、foreach 循环语句等，各个流程控制语句之间都可以嵌套使用。通过这种嵌套方法可以使程序更加灵活、实用。有关这种嵌套方法的实例可以参见前面的内容，几乎每个流程控制语句的实例中都应用到这种嵌套方法，这里不再赘述。

4.5　包含语句

引用外部文件可以减少代码的重用性，是 PHP 编程的重要技巧。PHP 提供了 4 个非常简单却很有用的包含语句，它们允许重新使用任何类型的代码。使用任意一个语句均可将一个文件载入到 PHP 脚本中，为网站创建统一风格的页面，从而更加轻松地编写和维护代码。



4.5.1　include()语句

使用 include()语句包含外部文件时，只有代码执行到该语句时才将外部文件包含进来，当所包含的外部文件发生错误时，系统只给出一个警告，而整个 PHP 文件则继续向下执行。其语法如下：

```
void include(string filename);
```

参数 filename 是指定的完整路径文件名。

例 4.09　如果 Web 页面具有一致的外观，可以在 PHP 中使用 include()语句将模板和标准元素载入到页面中。（**实例位置：光盘\mr\04\sl\4.09**）

具体步骤如下：

（1）创建一个简单 PHP 页，命名为 index.php。

（2）应用 include ()语句嵌入 3 个外部文件，分别是 job1.php 页、job2.php 页和 job3.php 页，代码如下：

```
<table id="__01" width="800" height="632" border="0" cellpadding="0" cellspacing="0">
	<tr>
		<td colspan="2">
			<?php include("job1.php");?>			//include 包含语句包含指定文件
		</td>
	</tr>
	<tr>
		<td>
			<?php include("job2.php");?>
		</td>
		<td>
			<?php include("job3.php");?>
		</td>
	</tr>
</table>
```

运行结果如图 4.23 所示。

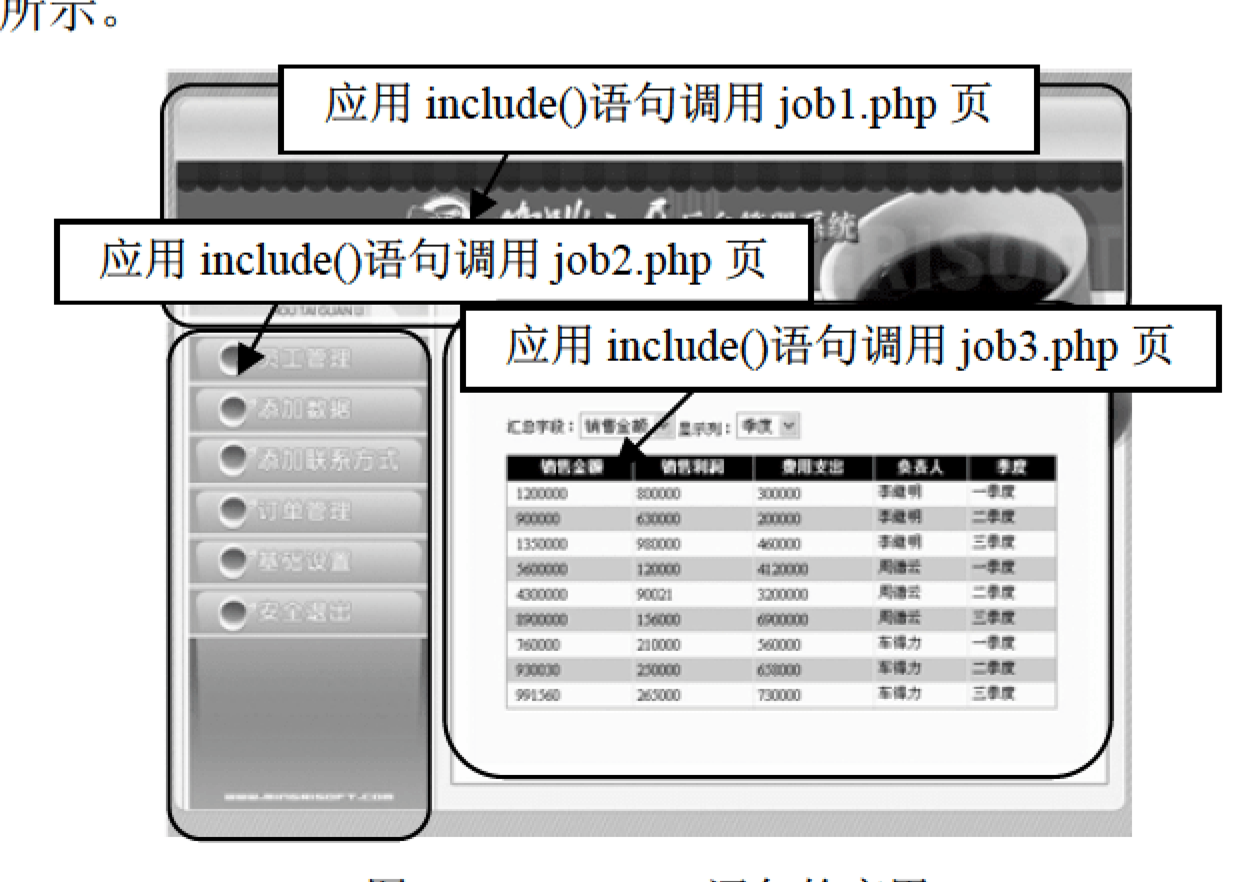


图 4.23　include()语句的应用

4.5.2　require()语句

require()语句与 include()语句类似，都是实现对外部文件的调用。其语法如下：

```
void require(string filename);
```

参数 filename 是指定的完整路径文件名。

当使用 require()语句载入文件时，它会作为 PHP 文件的一部分被执行。例如，通过 require()语句载入一个 mr.html 网页文件，那么文件内的任何 PHP 命令都会被处理。但是，如果将 PHP 脚本单纯地放到 HTML 网页中，它是不会被处理的。

通过上面的分析可以看出，PHP 可以使用任何扩展名来命名包含文件，如.inc 文件、html 文件或其他非标准的扩展名文件等。但 PHP 通常用来解析扩展名被定义为.php 文件。建议读者使用标准的文件扩展名。

例 4.10 下面应用 require()语句包含并运行指定的外部文件 job.php，具体代码如下。（**实例位置：光盘\mr\04\sl\4.10**）

```
<table width="974" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td><?php require("job.php");?></td>                    //require()语句包含指定文件
  </tr>
</table>
```

job.php 文件的代码如下：

```
<img src="images/top.jpg">
```

运行 index.php 页，输出 require()语句调用的 Web 页面。运行结果如图 4.24 所示。



图 4.24　使用 require()语句包含文件

4.5.3　include_once()语句

在使用 include_once()函数时，应该明确它与 include()函数的区别，应用 include_once()函数多次调用相同的文件时，程序只会调用一次。include_once()函数是 include()函数的延伸，它的作用和 include()函数几乎是相同的，唯一的区别在于 include_once()函数会在导入文件前先检测该文件是否在该页面的其他部分被导入过，如果有的话就不会重复导入该文件，这种区别是非常重要的。例如，要导入的文件中存在一些自定义函数，那么如果在同一个程序中重复导入这个文件，在第二次导入时便会发生错误，因为 PHP 不允许相同名称的函数被重复声明第二次。include_once()函数的语法如下：

```
void include_once (string filename);
```

参数 filename 是指定的完整路径文件名。

> 注意：何时应用 include()语句或 include_once()语句，应该根据不同的程序而定，但是建议使用 include_once()语句，因为应用 include_once()语句不会出现因重复包含了两个相同的文件而发生错误。

例 4.11 下面应用 include_once()语句包含并运行指定的外部文件 job.php，代码如下。（**实例位置：光盘\mr\04\sl\4.11**）

```
<table width="974" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td><?php include_once("job.php");?></td>            //include_once 语句包含指定文件
  </tr>
</table>
```

job.php 文件的代码如下：

```
<img src="images/top.jpg">
```

运行结果如图 4.25 所示。



图 4.25　include_once()语句的应用

4.5.4　require_once()语句

require_once()语句是 require()语句的延伸，它的功能与 require()语句基本类似，不同的是，require_once()语句会先检查要导入的文件是不是已经在该程序中的其他地方被调用过，如果有的话就不会再次重复调用该文件。例如，同时应用 require_once()语句在同一页面中调用了两个相同的文件，那么在输出时只有第一个文件被输出，第二次调用的文件不会被输出。require_once()语句的语法如下：

```
void require_once (string filename);
```

参数 filename 是指定的完整路径文件名。

例 4.12　下面应用 require_once()语句调用外部文件。**（实例位置：光盘\mr\04\sl\4.12）**

具体步骤如下：

首先，创建一个动态 PHP 文件，命名为 index.php，然后应用 require_once()函数嵌入 3 个外部文件，代码如下：

```
<table id="__01" width="800" height="632" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td colspan="2">
            <?php require_once("job1.php");?>        //require_once()语句包含文件
        </td>
    </tr>
    <tr>
        <td>
            <?php require_once("job2.php");?>
        </td>
        <td>
            <?php require_once("job3.php");?>
        </td>
    </tr>
</table>
```

本实例的运行结果如图 4.26 所示。

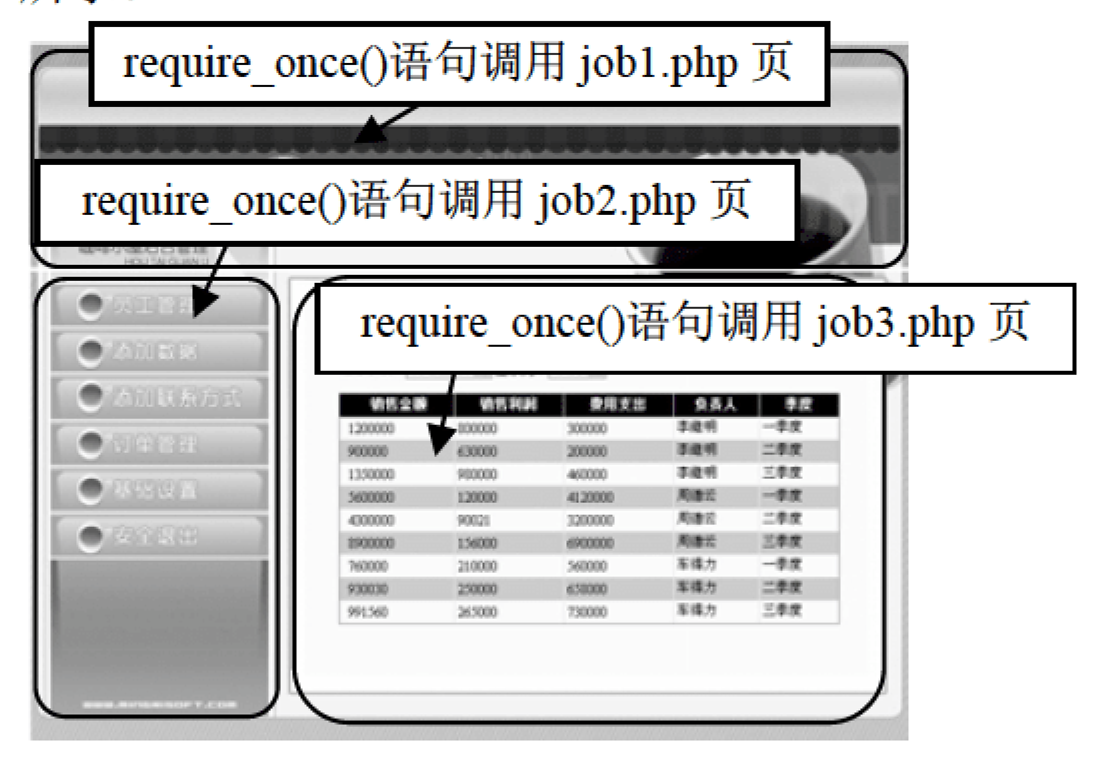


图 4.26　require_once()语句的应用

4.5.5　include()语句和 require()语句的区别

应用 require()语句来调用文件，其应用方法和 include()语句类似，但还存在一定的区别，区别如下：

- ☑ 在使用 require()语句调用文件时，如果没有找到文件，require()语句会输出错误信息，并且立即终止脚本的处理。而 include()语句在没有找到文件时则会输出警告，不会终止脚本的处理。
- ☑ 使用 require()语句调用文件时，只要程序一执行，会立刻调用外部文件；而通过 include()语句调用外部文件时，只有程序执行到该语句时，才会调用外部文件。

例 4.13　为了更好地区分 require()语句和 include()语句，下面通过具体的实例进行介绍。（**实例位置：光盘\mr\04\sl\4.13**）

（1）建立一个文件 index.php，分别使用 require()语句和 include()语句调用一个不存在的文件 topp.php。

（2）在 index.php 文件中应用 require()语句包含文件，并且指定错误的包含文件“topp.php”。程序代码如下：

```
<?php
echo "明日公司温馨提示：体会 require()语句的执行过程";
require("topp.php");                    //调用错误的外部文件，以查看 require()语句的执行效果
echo "上面的文件名错了，因此弹出错误提示，并且程序终止执行！";
?>
```

注意：上面是为了看出 require()语句在脚本中的执行效果，将正确的 top.php 文件名称写成错误的 topp.php 文件名。

当运行到 require()语句时，程序调用了不存在的 topp.php 文件，因此弹出错误信息，并终止程序的运行，没有输出最后一行字符串。运行结果如图 4.27 所示。

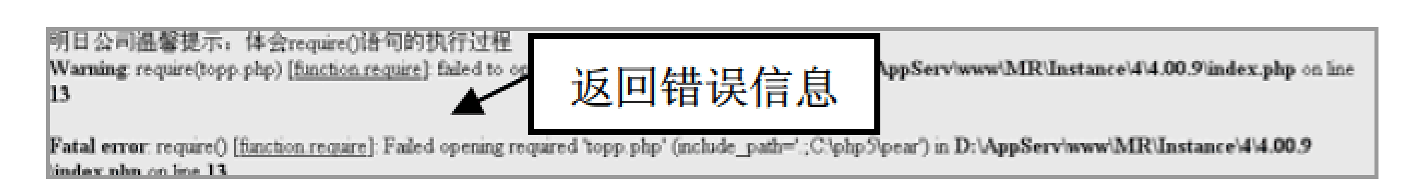


图 4.27　使用 require()语句包含错误的文件

注意：由于 require()语句调用了错误的外部文件，因此上面代码段中的“上面的文件名错了，因此弹出错误提示，并且程序终止执行！”字符串未被执行。

（3）在 include.php 文件中应用 include()语句包含外部文件，同样指定错误的文件名称 topp.php。程序代码如下：

```
<?php
echo "明日公司温馨提示：体会 include()语句的执行过程";
include("topp.php");                  //调用错误的外部文件，以查看 include()语句的执行效果
echo "哈哈，虽然上面的文件名错了，但是程序还是会继续执行！我被输出了！";
?>
```

当运行到 include()语句时，由于程序调用了不存在的 topp.php 文件，因此弹出警告信息。但与 require()语句不同的是，该程序没有终止执行，继续处理脚本文件，直到输出最后一行字符串为止。运行结果如图 4.28 所示。

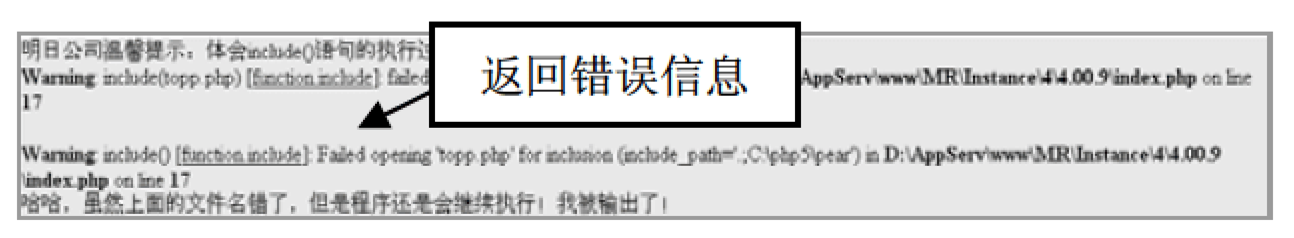
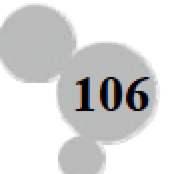


图 4.28　使用 include()语句包含错误的文件



你问我答：include_once()语句和 require_once()语句的区别？

include_once()语句和 require_once()语句的用途是确保一个被包含文件只能被包含一次。使用这两个语句可以防止意外地多次包含相同的函数库而导致函数的重复定义并产生错误的情况。但两者之间是有区别的，include_once()语句在脚本执行期间调用外部文件发生错误时产生一个警告，而 require_once()语句则导致一个致命错误。

4.6　照猫画虎——基本功训练

4.6.1　基本功训练 1——对与错的判断

视频讲解：光盘\mr\04\lx\对与错的判断.exe

实例位置：光盘\mr\04\zmhh\01

本实例是对变量$a 和变量$b 进行判断，当变量$a 的值大于变量$b 的值时，显示“明日科技”；如果变量$a 的值小于变量$b 的值时，显示“编程词典”。程序代码如下：

```
<?php
	$a=10;						//定义一个整型变量$a，值为 10
	$b=20;						//定义一个整型变量$b，值为 20
	if($a>$b){					//使用 if 语句判断$a 和$b 的大小
		echo"明日科技<br>";		//输出判断的结果对应的字符串
	}else{
		echo"编程词典<br>";
	}
?>
```

运行结果为：编程词典

照猫画虎：应用 if…else 语句判断随机数的奇偶性。（20 分）（实例位置：光盘\mr\04\zmhh\01_zmhh）

4.6.2　基本功训练 2——for 语句计算阶乘

视频讲解：光盘\mr\04\lx\for 语句计算阶乘.exe

实例位置：光盘\mr\04\zmhh\02

应用 for 循环计算 100 的阶乘，代码如下：

```
<?php
	$sum = 1;					//声明整型变量$sum
	for ($i = 1;$i <=100;$i++){
		$sum *= $i;				//当$i≤100 时，执行该表达式
	}
	echo "100! = ".$sum;		//输出该表达式
?>
```

运行结果为：100! = 9.33262144439E+147

在 for 语句中无论采用循环变量递增或递减的方式，有一个前提，就是一定要保证循环能够结束，无期限的循环（死循环）将导致程序的崩溃。在本例中是循环变量自增，循环的条件是$i≤100，由于在每次循环后$i 的值会加 1，当$i 的值大于 100 时，循环就结束了。

照猫画虎：以倒序的方式计算 100 的阶乘。提示：以循环变量递减的方式编写。（20 分）（实例位置：光盘\mr\04\zmhh\02_zmhh）

4.6.3 基本功训练 3——if 语句判断闰年

视频讲解：光盘\mr\04\lx\if 语句判断闰年.exe

实例位置：光盘\mr\04\zmhh\03

本实例应用 if 语句判断指定的年份是否为闰年，实例代码如下：

```
<?php
    $num = 2012;                          //定义一个变量$num，并为其赋值为 2012
    if(($num%4)==0 && ($num%100)!=0){     //如果变量$num 能够被 4 整除，而同时不能被 100 整除，则执行
下面的语句
        echo "$num".'年'."是闰年";         //输出是闰年
    }else{                                //否则输出是平年
        echo "$num".'年'."是平年";
    }
?>
```

运行结果为：2012 年是闰年

照猫画虎：判断今天是这个月的上、中或下旬。（20 分）（实例位置：光盘\mr\04\zmhh\03_zmhh）

4.6.4 基本功训练 4——用 switch 语句计算商品的折扣价格

视频讲解：光盘\mr\04\lx\用 switch 语句计算商品的折扣价格.exe

实例位置：光盘\mr\04\zmhh\04

本实例应用 switch 语句输出商品的折扣价格，运行结果如图 4.29 所示。选择要查询的商品，单击“提交”按钮，即可弹出电脑折扣价提示窗口，如图 4.30 所示。



图 4.29 商品查询



图 4.30 打折后的商品价格输出

具体步骤如下：

（1）新建一个 PHP 动态页，存储为 index.php。

（2）添加一个 form 表单、一个列表框、一个提交按钮，并应用表格对表单元素进行合理的布局。表单主要代码如下：

```
<form id="form1" name="form1" method="post" action="index.php">
  <table width="478" border="1" cellspacing="1" cellpadding="1">
    <tr>
      <td width="470" height="214" background="images/bg3.jpg"> <div align="center">
```



```
                <select name="select">
                  <option value="4000" selected="selected">电脑</option>
                  <option value="1000">MP4</option>
                  <option value="230">手机</option>
</select>
                <input type="submit" name="Submit" value="提交" />
</p>
          </div></td>
        </tr>
    </table>
</form>
```

（3）当用户在列表框中选择要查询的商品，并单击“提交”后，会弹出商品打折后的价格。关键代码如下：

```
<?php
if(isset($_POST['Submit']) && $_POST['Submit']=="提交"){          //通过 if 语句对提交按钮进行判断
      switch($_POST['select']){                              //应用 switch 语句通过$_POST 函数调用列表框的值
            case "230":                                         //输出商品的原价
                  $price=$_POST['select']*0.84;                 //计算商品打折价格
                  echo "<script>alert('手机折扣价为:$price');</script>"; //输出打折后的商品价格
            break;                                              //跳出循环
            case "4000":
                  $price=$_POST['select']*0.84;
                  echo "<script>alert('电脑折扣价为:$price');</script>";
            break;
            case "1000":
                  $price=$_POST['select']*0.84;
                  echo "<script>alert('Mp4 的折扣价为:$price');</script>";
            break;
      }
}
?>
```

照猫画虎：应用 switch 语句判断今天是星期几。（20 分）（实例位置：光盘\mr\04\zmhh\04_zmhh）

4.6.5　基本功训练 5——向一个循环语句中发布终止指令

视频讲解：光盘\mr\04\lx\向一个循环语句中发布终止指令.exe

实例位置：光盘\mr\04\zmhh\05

使用一个 break 语句，跳出 while 循环语句的输出。在 while 程序块中将声明一个随机数变量$tmp，当生成的随机数等于 10 时，使用 break 语句跳出循环，运行结果如图 4.31 所示。



图 4.31　使用 break 语句跳出循环

实例代码如下：

```
<?php
while(true){                              //使用 while 循环
    $tmp = rand(1,20);                    //声明一个随机数变量$tmp
    echo $tmp." ";                        //输出随机数
    if($tmp == 10){                       //判断随机数是否等于 10
        echo "<p>变量等于 10，终止循环";
        break;                            //如果等于 10，使用 break 语句跳出循环
    }
}
?>
```

说明：由于每次运行时，选取的随机数不同，循环输出的随机数也是不同的。

照猫画虎：使用 continue 语句跳出循环。（20 分）（实例位置：光盘\mr\04\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

4.7　情景应用——拓展与实践

4.7.1　情景应用 1——使用 if 语句判断美女征婚条件

视频讲解：光盘\mr\04\lx\使用 if 语句判断美女征婚条件.exe

实例位置：光盘\mr\04\qjyy\01

条件语句的应用，小到一个简单的判断登录用户权限的模块，大到高负载、高访问量的大型网站的整个程序中都会有它的身影，如何良好地掌握 if 条件语句的应用呢？其实很简单，每个应用语句都有其特点和使用原则，特别是在多层嵌套的 if 语句中它的应用原则是，无论多少层语句的嵌套，只要找到 if 语句开始处和闭合处一般就不会产生逻辑错误。

本应用的主要内容来源于网络上的一个笑话，其主要内容为一女子让电脑为其征婚，开出征婚条件有两点：一是要帅，二是要有车。电脑为她搜寻到了结果回答她说：女士您搜寻的是象棋吗？这位女子不愿接受此次搜寻的结果又重新输入条件，一是要有漂亮的房子，二是要有很多钱。电脑为她再次搜寻了结果：女士，您搜寻的是银行吗？此女子仍然不崩溃，继续输入条件，一是要长得酷，二是要有安全感，结果电脑搜出的是：女士，您搜寻的是奥特曼吗？此女子想尝试最后一次，于是输入条件，一是要帅，二是要有车，三是要有漂亮的房子，四是要有很多钱，五是要长得酷，六是有安全感。电脑搜寻了很久回答道：哦，女士，我知道了，您搜寻的是奥特曼在银行里下象棋。程序的运行结果如图 4.32 所示。

具体步骤如下：

（1）自定义函数，通过定义页面的布局，动态地向页面中插入文本和提交按钮的 value 属性值。其具体方法是，首先定义两个<div>标签，在内层的<div>标签中定义<p>标签，利用插入 PHP 代码的形式动态接收自定义函数首部$str 参数。提交按钮<input>标签中在 value 属性值的位置也使用同样的方法接收参数。具体代码如下：

```
<?php
    function if_isset($str,$str1){                                     //自定义函数
?>
    <div class="one">                                                  //div 标签
            <div id="one">                                             //div 标签
                //插入 PHP 代码，其输出内容为自定义函数的参数
<p><?php echo $str;?></p>
                <form action=""method="post">                          //定义表单
                    <input type="text" name="text" size="40" value="有车" >
                    <input type="text" name="text1"size="40" value="要帅" >
                    <input class="two" type="submit" name="<?php echo $str1?>" value="">
                </form>
            </div>
    </div>
<?php
    }
//当页面中不存在$_POST 变量时程序向下执行
    if(!isset($_POST[sub])&&!isset($_POST[sub1])&&!isset($_POST[sub2])){
        if_isset("一女子让计算机帮她搜寻结果，开出征婚条件有两点:","sub");//向自定义函数中传递参数
    }
                //当页面中存在变量$_POST[sub]时程序向下执行
    if(isset($_POST[sub])){
        if_isset("这位女子，不服搜出的结果又输入:","sub1");
    }
                //当页面中存在变量$_POST[sub1]时程序向下执行
    if(isset($_POST[sub1])){
        if_isset("此女子仍然不失望，继续输入条件:","sub2");
    }
                //当页面中存在变量$_POST[sub2]时程序向下执行
    if(isset($_POST[sub2])){
        if_isset("此女子仍然不失望，还继续输入条件:","sub3");
    }
?>
```



图 4.32　if 语句判断美女征婚条件

（2）利用 if…else 语句和 if…elseif 语句嵌套的形式，根据接收文本框参数的不同，输出对应的页面信息。代码如下：

```
<?php
/*
		1. 本代码中内嵌了 JavaScript 代码，其中 alert()函数是通过提示框向页面中打印信息
		2. location.href 语句的主要作用是执行页面跳转
*/
//当页面中存在$_POST 变量时程序向下执行
if(isset($_POST[sub])||isset($_POST[sub1])||isset($_POST[sub2])||isset($_POST[sub3])){
		//当获取的文本框内容 text 等于有车、text1 等于要帅两个条件同时满足时程序向下执行
		if($_POST[text]=="有车"&&$_POST[text1]=="要帅"){
			echo "<script>alert('计算机得出结果：女士，您搜寻的是象棋吗？')</script>";
		}elseif($_POST[text]=="要有漂亮的房子"&&$_POST[text1]=="要有很多钱"){
			echo "<script>alert('计算机得出结果：女士，您搜寻的是银行吗？')</script>";
		}elseif($_POST[text]=="要长得酷"&&$_POST[text1]=="要有安全感"){
			echo "<script>alert('计算机得出结果：女士，您搜寻的是奥特曼吗？')</script>";
		}elseif($_POST[text]=="有车，要帅，要有漂亮的房子"&&$_POST[text1]=="要有很多钱，要长得酷，
要有安全感"){
			echo "<script>alert('计算机得出结果：哦，女士，我知道了，您搜寻的是奥特曼在银行里下象棋');
location.href='index.php'</script>";
		}else{
		//当文本框所有内容都不匹配时，执行此段代码
			echo "<script>alert('输入信息错误');</script>";
		}
	}
?>
```

DIY：判断一位 43 岁的男士的退休年龄。提示：可以使用 if 语句和 if…else 语句（男士退休年龄是 60 岁，女士是 44 岁）。（20 分）（实例位置：光盘\mr\04\qjyy\01_diy）

4.7.2 情景应用 2——使用 switch 语句定制网页每日问候语

视频讲解：光盘\mr\04\lx\使用 switch 语句定制网页每日问候语.exe

实例位置：光盘\mr\04\qjyy\02

通过 switch 语句判断当前日期，并且根据当前日期给出提示信息。运行结果如图 4.33 所示。



图 4.33 输出贴心提醒警句

使用 switch 语句来输出当天为星期几，并根据不同的日期输出不同的贴心提醒警句，代码如下：

```
<?php
setlocale(LC_TIME,"chs");                   //设置本地环境
$weekday = strftime("%A");                  //声明变量$weekday，通过 strftime()函数获取今天是星期几
switch ($weekday){                          //switch 语句，判断$weekday 的值
```

```
    case "星期一":                          //如果变量的值为“星期一”
        echo "今天是".iconv('gbk','utf-8',$weekday)."，新的一周开始了！";
    break;
    case "星期二":                          //如果变量的值为“星期二”
        echo "今天是".iconv('gbk','utf-8',$weekday)."，时刻保持良好的工作状态!";
    break;
    case "星期三":                          //如果变量的值为“星期三”
        echo "今天是".iconv('gbk','utf-8',$weekday)."，劳动者是最美的人，努力工作哟！";
    break;
    case "星期四":                          //如果变量的值为“星期四”
        echo "今天是".iconv('gbk','utf-8',$weekday)."，勤奋才能创造绩效，加油！）";
    break;
    case "星期五":                          //如果变量的值为“星期五”
        echo "今天是".iconv('gbk','utf-8',$weekday)."，一定要出色地完成本周工作哟！";
    break;
    case "星期六":                          //如果变量的值为“星期六”
        echo "今天是".iconv('gbk','utf-8',$weekday)."，可以睡到自然醒！";
    break;
    default:                                //默认值
        echo "今天是".iconv('gbk','utf-8',$weekday)."，呵呵，轻松地玩上一天！";
    break;
}
?>
```

DIY：应用 switch 语句输出本年度当前月份的活动安排。（20 分）（实例位置：光盘\mr\04\qjyy\02_diy）

4.7.3　情景应用 3——使用 while 语句循环读取数组中数据

视频讲解：光盘\mr\04\lx\使用 while 语句循环读取数组中数据.exe

实例位置：光盘\mr\04\qjyy\03

while 循环语句常与查询数据库的返回结果一起应用，循环输出查询后的每条结果。本实例应用数组模拟一个数据库查询图书信息后返回结果，利用 list…each 语句与 while 语句配合，循环输出每条商品的信息。运行结果如图 4.34 所示。



ID	商品名称	价 格	数量	金额
0	品牌笔记本	4998元	数量	1
1	高档男式衬衫	588元	1	588
2	品牌3G手机	4666元	2	9332
3	高档女士挎包	698元	1	698

图 4.34　while 循环输出商品信息

首先，声明一个二维数组，代码如下：

```
<?php
$sp=array(                                  //声明一个二维数组
```

```
            array("品牌笔记本","4998 元","数量","1","4998"),
            array("高档男式衬衫","588 元","1","588"),
            array("品牌 3G 手机","4666 元","2","9332"),
            array("高档女士挎包","698 元","1","698")
      );
?>
```

其次，创建一个输出商品内容的表格。

最后，利用 list…each 语句与 while 语句配合，循环输出每条商品的信息，关键代码如下：

```
<?php
while(list($key, $value)=each($sp)){          //利用 list…each 语句输出键和值
      echo '<tr><td height="24" align="center" bgcolor="#FFFFFF" class="STYLE2">'.$key.'
                  </td><td align="center" bgcolor="#FFFFFF" class="STYLE2">'.$value[0].'
                    </td><td align="center" bgcolor="#FFFFFF" class="STYLE2">'.$value[1].'
                    </td><td align="center" bgcolor="#FFFFFF" class="STYLE2">'.$value[2].'
                    </td><td align="center" bgcolor="#FFFFFF" class="STYLE2">'.$value[3].'
                    </td></tr>';
}
echo '</table>';
?>
```

DIY：应用 foreach 语句输出数组中的数据。（20 分）（实例位置：光盘\mr\04\qjyy\03_diy）

4.7.4 情景应用 4——switch 网页框架

视频讲解：光盘\mr\04\lx\switch 网页框架.exe

实例位置：光盘\mr\04\qjyy\04

网页框架是 PHP 脚本编程中很重要的一部分，好的框架制作手法相当于一件了不起的工艺品，是一种逻辑思维上的提升。通过 switch 语句实现一个简单的框架功能，即在网站的主页面中，单击不同的超链接分别调用不同的网页显示于主页面中。充分发挥超链接中栏目标识的作用，将栏目标识作为 switch 语句表达式的值，根据不同的栏目标识调用不同的网页。运行结果如图 4.35 所示。



图 4.35　应用 switch 语句实现框架的功能

首先，判断当页面中存在$_GET['id']的值时，将$_GET['id']的值保存在变量$string 中。通过 switch 语句根据接收到的参数不同包含不同的页面，关键代码如下：

```
<?php
if(isset($_GET['id'])){
      if(isset($_GET['id'])||$_GET['id']==NULL){          //判断接收到的参数是否存在，如果存在向下执行
            $string=$_GET[id];                              //定义变量将$_GET[id]保存到$string 中
```



```
                switch($string){                                    //switch 语句结构
                    case "0";                                       //值为 0 时程序执行对应代码
                    include("bg.html");                             //包含页面
                    break;                                          //跳出 switch 语句
                    case "1";                                       //值为 1 时程序执行对应代码
                    echo"<img src='images/bg_03.jpg'>";             //输出图像
                    break;                                          //跳出 switch 语句
                    case NULL;                                      //当为空值时执行对应代码
                    echo"<img src='images/bg_03.jpg'>";             //输出图像
                    break;                                          //跳出 switch 语句
                    }
            }
    }
    ?>
```

DIY：应用 switch 语句设计网站主页。（20 分）（实例位置：光盘\mr\04\qjyy\04_diy）

4.7.5　情景应用 5——使用 for 循环语句实现多图片上传

视频讲解：光盘\mr\04\lx\使用 for 循环语句实现多图片上传.exe

实例位置：光盘\mr\04\qjyy\05

上传图片是很多娱乐网站所必须包含的模块，其优点是良好的与用户互动，由用户选择喜欢的图片或者文件上传。大量的用户为网站传递数以千计的文件和图片，这对于一个企业网站来说，是一个相当可观的收益。单文件上传实现起来相当简单，这里使用 for 循环语句，为图片上传增加了一些扩展——多文件上传。本实例应用 for 循环控制语句实现将多图片上传到服务器指定文件夹下，一次可以上传 4 张以内的图片。运行结果如图 4.36 所示。



图 4.36　for 循环控制语句的应用

开发本实例需要对文件系统的知识有初步的了解，其中应用到$_FILE 数组和 move_upload_file()函数，有关其详细讲解请读者参考第 13 堂课。

首先，创建 form 表单，添加 4 个文件域和提交按钮，将数据通过 POST 方法提交到指定的位置。

然后，通过$_POST[]全局数组获取表单中提交的数据，应用 for 循环获取表单中提交的数据，通过$_FILES 全局数组和 move_upload_file()函数完成文件的上传操作。关键代码如下：

```
<?php
    if(isset($_POST[sub])){                         //如果页面中存在变量$_POST[sub]程序向下执行
        for($a=1;$a<4;$a++){                        //for 循环语句
            $name=$_FILES['file_'.$a];              //将图片信息保存在变量中
```

```
                    $rand=rand(1,100000);                         //获取 1～100000 的随机数
                    $name_type=substr($name['name'],-4,4);        //截取图片名称的后 4 位数据并保存在变量中
                    $time=microtime();                            //获取时间戳微秒数
                    $path='upfiles/'.(($rand+$time).$name_type);  //上传路径
                    $dir='upfiles/';                              //文件夹名称
                    $move = move_uploaded_file($name['tmp_name'],$path);      //将图片移动到指定文件夹下
                    if($move==true){
                        echo "成功上传文件".($rand+$time).$name_type."<br>";
                    }
                }
            }
?>
```

DIY：应用 for 语句开发一个乘法口诀表。提示：利用 for 循环语句将计算结果输出在特定的表格中。（20 分）（实例位置：光盘\mr\04\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

4.8 自 我 测 试

一、选择题（每题 10 分，5 道题）

1．以下代码的运行结果为（　　）。

```
<?php
    if($i=""){
        echo "a";
    }else{
        echo "b";
    }
?>
```

A．输出 a　　B．输出 b　　C．条件不足，无法确定　　D．运行出错

2．哪一种语言结构最能代表以下多路分支 if 语句的选择结构（　　）。

```
<?php
    if($a=='a'){
        somefunction();
        }else if($a=='b'){
            anotherfunction();
        }else if($a=='c'){
            dosomething();
        }else{
            donothing();
    }
?>
```

A．没有 default 的 switch 语句　　B．一个递归的函数调用

C．一个 while 语句　　D．它是唯一的逻辑结构　　E．使用 default 的 switch 语句

3．以下代码返回的结果为（　　）。

```
<?php
	function p(){
		return 1;
	}
	if(p()){
		echo"fasle";
		}else{
			echo"true";
		}
?>
```

A．true　　B．false　　C．程序运行出错　D．根据版本来定

4．以下程序的运行结果为（　　）。

```
<?php
	$var=FALSE;
	if(empty($var)){
		echo"null";
	}else{
		echo"have value";
	}
?>
```

A．null　　B．have value　　C．无法确定　　D．什么也不显示，提示错误

5．请指出下面的代码是否有问题，如果有问题，错误有几处？（　　）

①if test num1 –gt num2

②then

③echo “Test”

④elseif

⑤echo “test”

⑥fi

A．正确，该代码没有问题　　B．有两处错误　　C．有 3 处错误　　D．有 4 处错误

二、填空题（每题 10 分，5 道题）

1．语句 include 和 require 都能把另外一个文件包含到当前文件中，它们的区别是（　　　）；为了避免多次包含同一文件，可以用语句（　　　）和（　　　）来代替它们。

2．程序设计的结构大致可以分为（　　　）种，它们分别是（　　　）。

3．下面代码的运行结果为（　　　）。

```
<?php
	$sum=0;
	for ($i=1;$i<=10;$i++){
	if($i>5)break;
	$sum=$sum+$i;
	}
	echo"sum=".$sum;
?>
```

4．以下程序的运行结果为（　　　）。

```
<?php
	$op1=1;
	$sum=0;
	do{
	$sum=$sum+$op1;
	$op1=$op1+1;
	}while($op1<=10);
	echo"sum=".$sum;
?>
```

5．以下程序的运行结果为（　　　）。

```
	<?php
	$num=1;
	while($num!=1){
		echo"不会看到.";
	}
	do{
		echo"会看到。";
	}while($num!=1);
?>
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

4.9　行 动 指 南

开始日期：______年____月____日　　　　　　　　　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目 分数（　　）	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
		分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目 分数（　　）	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
		分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目 分数（　　）	分数>75 分	优秀，有成为编程高手的潜质。
		分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）使用循环语句输出任意一个二维数组。	
		（2）使用循环控制语句输出杨辉三角。	
		（3）应用 switch…case 分支控制语句输出一天的活动安排。	
		（4）使用 do…while 语句和 while 语句同时输出一个数组，看看两者的区别在哪里。	

续表

序号	内　　容	行 动 指 南
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

4.10　成功可以复制——BEA 创始人之一庄思浩

成就：1995 年与两位同事离开 SUN 公司创立 BEA。1997 年 BEA 公司在纳斯达克上市，成为全球最大中间件公司。

创始公司：BEA 公司

财富巅峰：资产范围 10 亿以上

出生日期：1962 年

最高学历：计算机硕士学位

超级语录：设定自己的目标，勇于超载自己，不要追随别人；要有恒心，不向困难低头，永不轻言放弃。

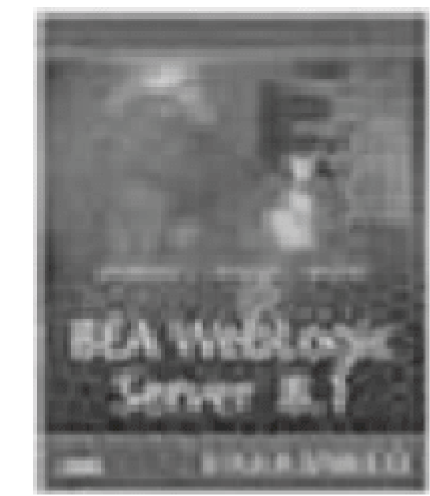
成功历程

庄思浩祖籍宁波，出生于香港，9 岁赴美。19 岁他就从旧金山的大学毕业，获得计算机学士学位。然后进入加州大学戴维斯分校，获得计算机硕士学位，专攻分布式数据管理。刚毕业时，庄思浩还十分犹豫；当初想走教授之路，那是 20 世纪 80 年代大多数华人学子的首选。他的朋友极力劝他加入 SUN，那时 SUN 也是个初创公司。不过事实证明，这个选择十分英明，庄思浩在 SUN 摸爬滚打了 9 年，收获颇丰。而 SUN 也发展成 2 万多人的公司，成为 IT 业的巨头之一。这时候，创业的萌芽开始萌动。

1994 年，曾经在 SUN 共事的好友比尔·科尔曼（Bill Coleman）来找他，鼓动他一起创立公司。两个人就在他父母的公寓里构思、讨论与设计。最初的目标是为企业开发和部署可靠的、可伸缩的业务应用提供完善的基础结构，最后选择了中间件作为主攻方向。不久，爱德·斯考特（ED Scott）也加盟，这是一位每个细胞都充满了销售的人物，是公司迅速获得收入的重量级武器。公司刚刚起步，没有风险资金，3 位创始人自掏腰包，并将公寓为据点。即使这样，也只够支撑 18 个月，庄思浩给自己立下军令状："奋斗 18 个月，做不成，就去思科（Cisco）打工。"1995 年 1 月公司成立，公司名字 BEA 就是 3 个人名字的字母组成。

这是一个梦幻组合。幸运女神也逐渐到来。这一年，网景公司上市，点燃了互联网的烈火，历史上最壮观的技术创新和最美好的创业浪潮开始了。

当年 5 月份，BEA 就融资 5000 万美元。在强大的资金催动下，收入开始火箭般跃升：第一年 900 万美元，第二年 6000 万美元，第三年 2 亿美元，2000 年达到 7 亿多美元，1999 年，BEA 股票入选德意志银行 e-Process 股票指数，随后入选高盛电子商务指数、《标准》100 股票指数、Nasdaq-100 指数、eCompany Now! 40 指数，再到入选《财富》e-50 指数，BEA 在业内的重要性的确是与日俱增。软件业的又一个奇迹就这样诞生了。



BEA 产品

当然，对于庄思浩，人们最关心、最想问的还是成功的秘诀，“最重要的是工作的狂热、创业路十分艰难，建设公司文化、管理人事、争取客户都是头疼连连而且总会出现各种考验，愿意撑过难关，不随意放弃，不随意将公司出售。”这是庄思浩自认为的成功之道。

深度评价

庄思浩的成功告诉我们，要设定自己的目标，用于超越自己，不要追随别人；做事要有恒心，不向困难低头，永不轻言放弃，因为在个人成功的路上会有各种各样的障碍，只有坚持才有机会成功；做人要保持一颗平常心，并且敢于正视自己的错误。

第 5 堂课

PHP 数组应用

（视频讲解：105 分钟）

数组提供了一种快速、方便地管理一组相关数据的方法，是 PHP 程序设计中的重要内容。通过数组可以对大量性质相同的数据进行存储、排序、插入及删除等操作，从而可以有效地提高程序开发效率及改善程序的编写方式。在本堂课中将对 PHP 中的数组操作技术进行系统、详细的讲解。

学习摘要：

- **数组概述**
- **数组类型**
- **声明数组**
- **遍历、输出数组**
- PHP **数组函数**
- PHP **全局数组**

5.1 数 组 概 述

数组是一组数据的集合，将数据按照一定规则组织起来，形成一个可操作的整体。是对大量数据进行有效组织和管理的手段之一，通过数组函数可以对大量性质相同的数据进行存储、排序、插入及删除等操作，从而可以有效地提高程序开发效率及改善程序的编写方式。

数组的本质是储存、管理和操作一组变量。数组与变量的比较效果如图 5.1 所示。

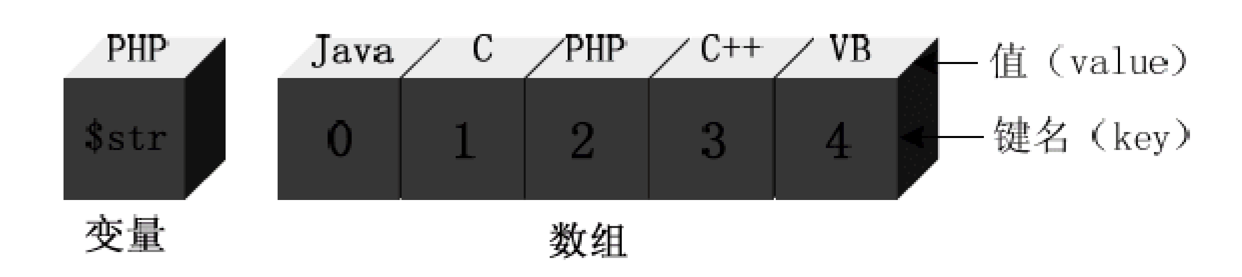


图 5.1　数组与变量的比较效果

5.2 数 组 类 型

PHP 中将数组分为一维数组、二维数组和多维数组，但无论是一维还是多维，可以统一将数组分为数字索引数组（indexed array）和关联数组（associative array）两种。数字索引数组使用数字作为键名（如图 5.1 中展示的就是一个数字索引数组），关联数组使用字符串作为键名（如图 5.2 所示）。

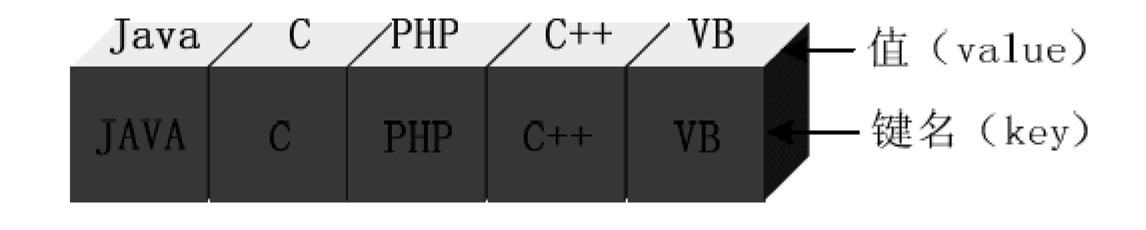


图 5.2　关联数组

（1）数字索引数组

数字索引数组，下标（键名）由数字组成，默认从 0 开始，每个数字对应数组元素在数组中的位置，不需要特别指定，PHP 会自动为数字索引数组的键名赋一个整数值，然后从这个值开始自动增量。当然，也可以指定从某个具体位置开始保存数据。

数组中的每个实体都包含键名和值两项。可以通过键名来获取相应数组元素（值），如果键名是数值，那么就是数字索引数组；如果键名是数值与字符串的混合，那么就是关联数组。

下面创建一个数字索引数组，代码如下：

```
$arr_int = array ("PHP24 堂课","C#24 堂课","VB24 堂课"); //声明数字索引数组
```

（2）关联数组

关联数组，下标（键名）由数值和字符串混合的形式组成。如果一个数组中有一个键名不是数字，那么这个数组就叫做关联数组。

关联数组（associative array）使用字符串键名来访问存储在数组中的值，如图 5.2 所示。

下面创建一个关联索引数组，代码如下：

```
$arr_string = array ("PHP"=>"PHP24 堂课","JAVA"=>"JAVA24 堂课","C#"=>"C#24 堂课"); //声明关联数组
```



说明： 关联数组的键名可以是任何一个整数或字符串。如果键名是一个字符串，则要给这个键名或索引加上一个定界修饰符——单引号（'）或双引号（"）。对于数字索引数组，为了避免不必要的麻烦，最好也加上定界符。

5.3　声明数组

在讲解数组的声明方法之前，先来了解一下数组的命名规则。PHP 中声明数组的规则如下：

（1）数组的名称由一个美元符号开始，第一个字符是字母或下划线，其后是任意数量的字母、数字或下划线。

（2）在同一个程序中，标量变量和数组变量都不能重名。例如，如果已经存在一个名称为$string 的变量，而又创建一个名称为$string 的数组，那么前一个变量就会被覆盖。

（3）数组的名称区分大小写，如$String 与$string 是不同的。

声明数组的方法有两种，分别为用户声明和函数声明。下面看一下用户如何自己创建数组和使用什么函数可以直接创建数组。

5.3.1　用户创建数组

用户创建数组应用的是标识符“[]”，通过标识符“[]”可以直接为数组元素赋值。其基本格式如下：

```
$arr[key] = value;
$arr['0'] = value;
```

其中 key 可以是 int 型或者字符串型数据，value 可以是任何值。

例 5.01　应用标识符“[]”创建数组 array，然后应用 print_r()函数输出数组元素，代码如下。**（实例位置：光盘\mr\05\sl\5.01）**

```
<?php
$array['0']="PHP24 堂课";          //通过标识符[]定义数组元素值
$array['1']="JAVA24 堂课";         //通过标识符[]定义数组元素值
$array['2']="VB24 堂课";           //通过标识符[]定义数组元素值
$array['3']="VC24 堂课";           //通过标识符[]定义数组元素值
print_r($array);                   //输出所创建数组的结构
?>
```

运行结果如图 5.3 所示。

本例中使用 print_r()函数输出数组元素，因为使用 print_r 输出数组，将会按照一定格式输出数组中所有的键名和元素。而使用 echo 语句可以输出数组中指定的某个元素，下面使用 echo 语句输出数组中的第一个元素，如图 5.4 所示。

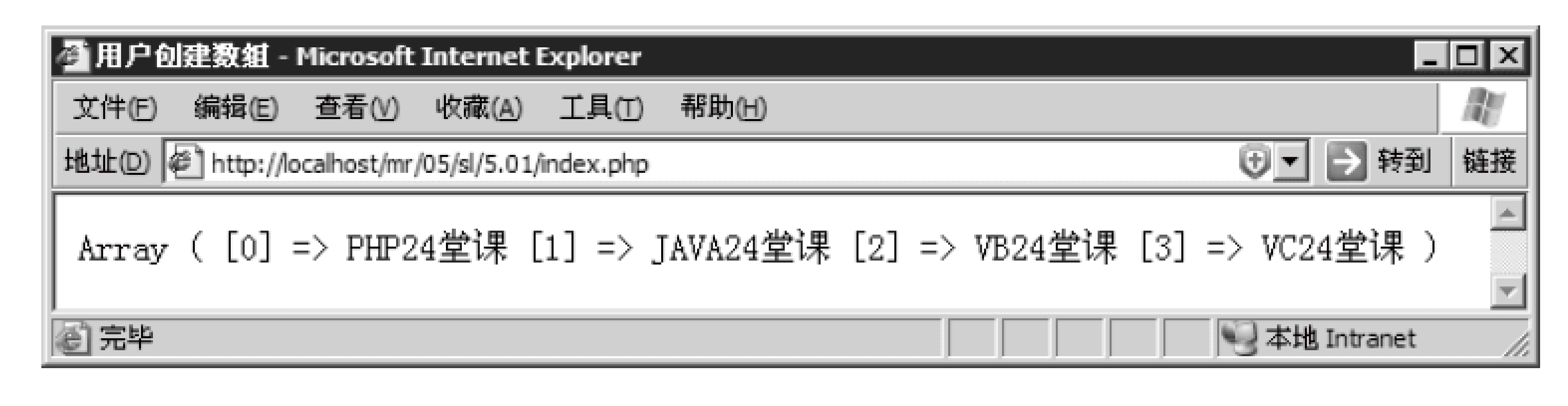


图 5.3　通过标识符[]创建的数组结构



图 5.4　使用 echo 输出数组

说明：（1）用户创建数组，比较适合创建未知大小的数组，或者创建大小可能发生改变的数组。

（2）切忌在通过标识符[]直接为数组元素赋值时，同一数组元素中的数组名称必须相同。

5.3.2 函数创建数组

PHP 中最常用的创建数组的函数是 array()，其语法如下：

```
array array ( [mixed ...])
```

参数 mixed 的格式为“key => value”，多个 mixed 参数之间用逗号分开，分别定义索引（key）和值（value）。

应用 array()函数声明数组时，数组下标既可以是数值索引，也可以是关联索引。下标与数组元素值之间用“=>”进行连接，不同数组元素之间用逗号进行分隔。

应用 array()函数定义数组时，可以在函数体中只给出数组元素值，而不必给出键名。

说明：（1）数组中的索引（key）可以是字符串或数字。如果省略了索引，会自动产生从 0 开始的整数索引。如果索引是整数，则下一个产生的索引将是目前最大的整数索引+1。如果定义了两个完全相同的索引，则后面一个会覆盖前一个。

（2）数组中的各数据元素的数据类型可以不同，也可以是数组类型。当 mixed 是数组类型时，就是二维数组。

例 5.02 应用 array()函数声明数组，并输出数组中的元素，代码如下。**（实例位置：光盘\mr\05\sl\5.02）**

```
<?php
$arr_string=array('one'=>'php','two'=>'java');          //以字符串作为数组索引，指定关键字
print_r($arr_string);                                   //通过 print_r()函数输出数组
echo "<br>";
echo $arr_string['one']."<br>";                         //输出数组中的索引为 Java 的元素
$arr_int=array('php','java');                           //以数字作为数组索引，从 0 开始，没有指定关键字
print_r($arr_int);                                      //输出整个数组
echo "<br>";
echo $arr_int['0']."<br>";                              //输出数组中的第一个元素
$arr_key=array(0 =>'PHP24 堂课', 1 =>'JAVA24 堂课', 1 =>'VB24 堂课');  //指定相同的索引
print_r($arr_key);                                      //输出整个数组，发现只有两个元素
?>
```

运行结果如图 5.5 所示。

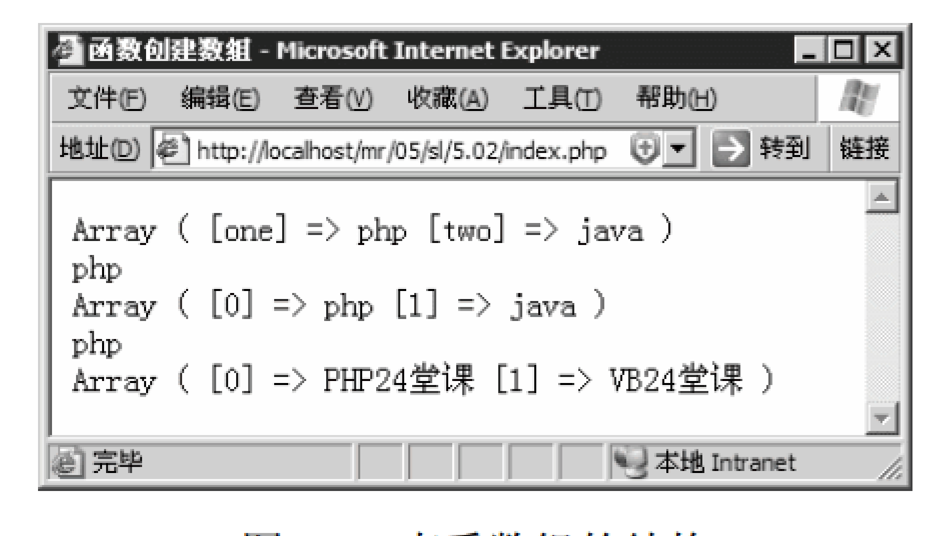


图 5.5 查看数组的结构

5.3.3 创建二维数组

上述创建的数组都是只有一列数据内容的，因此称为一维数组。如果将两个一维数组组合成一个数组，



那么就称为二维数组。

例 5.03 下面应用 array()函数创建一个二维数组，并输出数组的结构，代码如下。**（实例位置：光盘\mr\05\sl\5.03）**

```
<?php
$str = array (
      "网络编程图书"=>array ("PHP24 堂课","C#24 堂课","VB24 堂课"),
          "历史图书"=>array ("1"=>"春秋","2"=>"战国","3"=>"左传"),
          "文学图书"=>array ("明朝哪些事儿",3=>"狼图腾","鬼吹灯")
      );                        //声明二维数组
print_r ($str) ;                //输出数组元素
?>
```

运行本实例，查看运行结果的源文件，如图 5.6 所示。

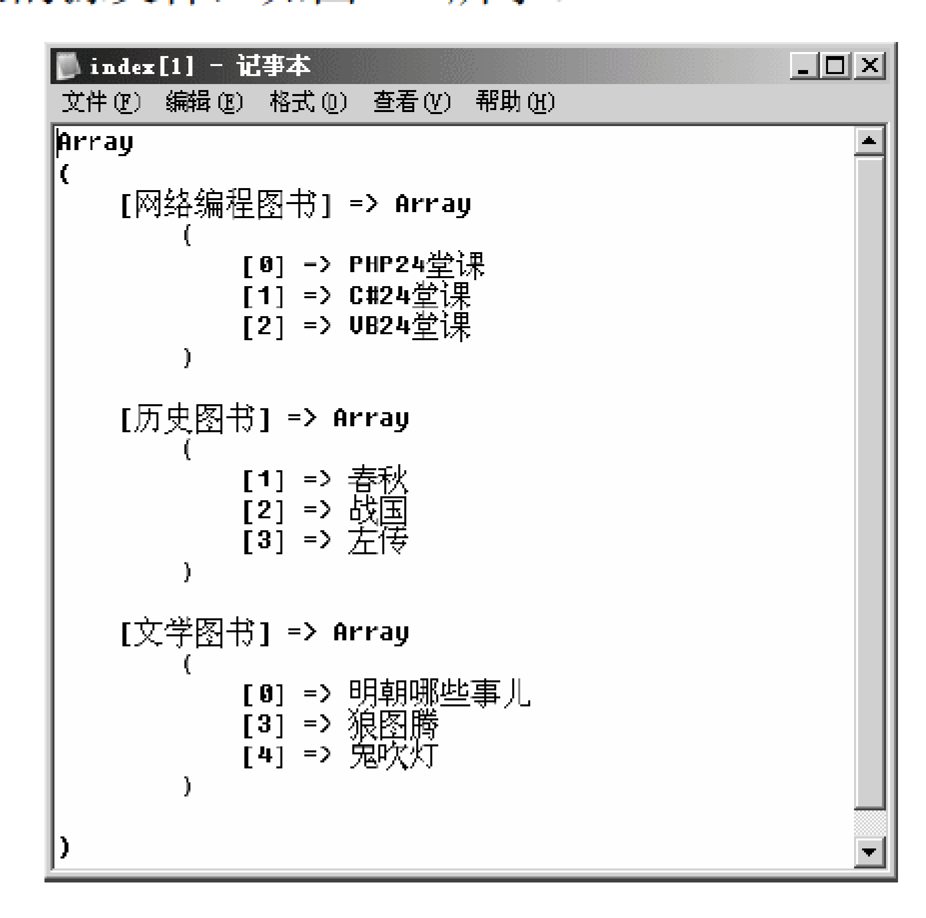


图 5.6　二维数组的结构

说明：根据创建二维数组的方法，举一反三，就可以很容易地创建三维、四维或多维数组。

5.4　遍历、输出数组

5.4.1　遍历数组

遍历数组就是按照一定的顺序依次访问数组中的每个元素，直到访问完为止。PHP 中可以通过流程语句（foreach 和 for 循环语句）和函数（list()和 each()）来遍历数组，下面分别介绍这几种遍历数组的方法。

1．foreach 语句遍历数组

在第 4 堂课中已经介绍了 foreach 语句的循环结构，下面使用该语句来遍历数组。

☑　使用 foreach 语句遍历一维数组

例 5.04　下面使用 foreach 语句遍历一维数组$str，具体代码如下。**（实例位置：光盘\mr\05\sl\5.04）**

```
<?php
//创建数组
$str=array('编程词典网'=>'www.mrbccd.com',
             '明日图书网'=>'www.mingribook.com',
```

```
);
echo "原数组：";
print_r($str);
echo "<br>";
echo "遍历后的值：";
foreach($str as $link){              //遍历数组
      echo $link."  ";
}
?>
```

运行结果如图 5.7 所示。

上面的例子中是将数组的值遍历输出，下面将数组的键名和元素值都遍历输出，只需将例 5.04 中的 foreach 循环语句改为：

```
foreach($str as $key=>$link){
      echo "$key----$link"."<br>";             //对应输出数组中的键名和元素值
}
```

运行结果如图 5.8 所示。



图 5.7　使用 foreach 语句遍历数组（遍历值）



图 5.8　使用 foreach 语句遍历数组（遍历键名和值）

☑　使用 foreach 语句遍历二维数组

例 5.05　上面是遍历一维数组的方法，如果想要遍历多维数组，该怎么办呢？方法很简单，只要使用两个 foreach 语句来处理即可，具体代码如下。**（实例位置：光盘\mr\05\sl\5.05）**

```
<?php
$str = array (
      "网络编程图书"=>array ("PHP24 堂课","C#24 堂课","VB24 堂课"),
            "历史图书"=>array ("1"=>"春秋","2"=>"战国","3"=>"左传"),
            "文学图书"=>array ("明朝哪些事儿",3=>"狼图腾","鬼吹灯")
      );                                    //二维数组
foreach($str as $key=>$link){           //使用 foreach 语句获取一维数组的元素和值
      foreach($link as $value){         //使用 foreach 语句获取上一个 foreach 语句遍历得到的元素和值中的值
            echo $value."<br>";
      }
}
?>
```

运行结果如图 5.9 所示。

2．for 语句遍历数组

如果要遍历的数组是数字索引数组，并且数组的索引值为连续的整数时，可以使用 for 循环来遍历，但前提条件是需要应用 count()函数获取到数组中元素的数量，然后将获取的元素数量作为 for 循环执行的条件，才能完成数组的遍历。

例 5.06　下面使用 for 循环来遍历数组$array，具体代码如下。**（实例位置：光盘\mr\05\sl\5.06）**



```
<?php
$array=array(                              //定义数组
                "0"=>"PHP24 堂课",
                "1"=>"JAVA24 堂课",
                "2"=>"VB24 堂课",
                "3"=>"VC24 堂课"
                );
for($i=0;$i<count($array);$i++){           //使用 for 循环遍历数组
     echo $array[$i]."<br>";               //输出数组元素
}
?>
```

运行结果如图 5.10 所示。

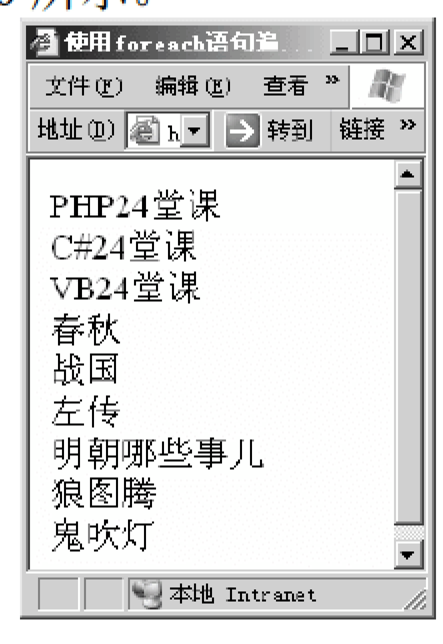


图 5.9　使用 foreach 语句遍历多维数组

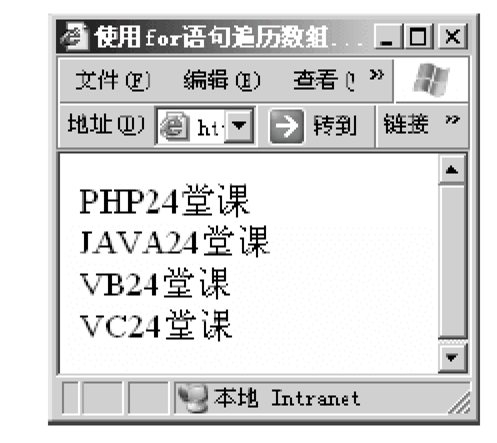


图 5.10　使用 for 循环遍历数组

3．通过数组函数 list()和 each()遍历数组

☑　list()函数

list()函数将数组中的值赋给一些变量，该函数仅能用于数字索引的数组，且数字索引从 0 开始。其语法如下：

```
void list ( mixed ...)
```

参数 mixed 为被赋值的变量名称。

☑　each()函数

each()函数返回数组中的键名和对应的值，并向前移动数组指针。其语法如下：

```
array each ( array array)
```

参数 array 为输入的数组。

例 5.07　下面使用 list()和 each()函数来遍历数组$array，具体代码如下。（**实例位置：光盘\mr\05\sl\5.07**）

```
<?php
$array=array(                              //定义数组
          "0"=>"PHP24 堂课",
          "1"=>"JAVA24 堂课",
          "2"=>"VB24 堂课",
          "3"=>"VC24 堂课"
          );
/*
使用 list()函数获取 each()函数中返回数组的值
并分别赋给$name 和$value，然后使用 while 循环输出
*/
while(list($name,$value)=each($array)){
```

```
    echo $name=$value."<br>";          //输出 list()函数获取到的键名和值
}
?>
```

运行结果如图 5.11 所示。

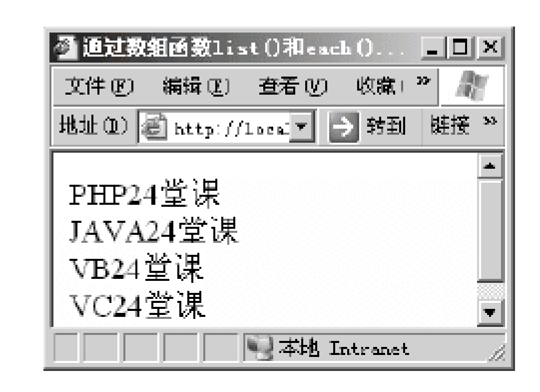


图 5.11　通过数组函数 list()和 each()遍历数组

5.4.2　输出数组元素

在前面已经实践过数组的输出，就是使用 print_r()函数和 echo 语句。

☑　利用 print_r()函数可以输出数组的结构，也可以使用 var_dump()函数，同样是输出数组的结构。

☑　利用 echo 语句则是单纯地输出数组中的某个元素，而且要有标识符[]和数组索引的配合，其格式是“echo $array[0]”。同样还有 print 语句，它也可以单纯地输出数组中的某个元素值。

5.5　PHP 数组函数

下面介绍一些在实际程序开发中比较常用的数组函数，如果读者需要使用数组函数实现某些特殊的功能，建议参考 PHP 中文手册，那里有对所有数组函数的详细介绍，可以根据所要实现的功能进行选择、学习或者研究。

5.5.1　统计数组元素个数

在 PHP 中，应用 count()函数可以对数组中的元素个数进行统计，在讲解使用 for 循环遍历数组时已经应用到，下面详细介绍该函数，其语法格式如下：

```
int count ( mixed var [, int mode])
```

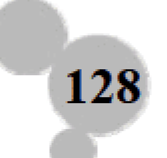
参数 var 指定操作的数组对象；参数 mode 为可选参数，如果 mode 的值设置为 COUNT_RECURSIVE（或 1），count()函数检测多维数组。参数 mode 的默认值为 0。该函数返回数组元素的个数。

说明： 如果 count()函数的操作对象是 null，那么返回结果是 0。count()函数对没有初始化的变量返回 0，但对于空的数组也会返回 0。如果要判断变量是否初始化，则可以应用 isset()函数。count()函数不能识别无限递归。

例 5.08　下面使用 count()函数统计数组中的元素个数，并输出统计结果，代码如下。**（实例位置：光盘\mr\05\sl\5.08）**

```
<?php
$array=array(0 =>'PHP24 堂课', 1 =>'JAVA24 堂课', 2 =>'VB24 堂课',3=>"VC24 堂课");
echo count($array);                //统计数组中元素个数，并使用 echo 语句输出统计结果
?>
```

运行结果为：4



5.5.2　向数组中添加元素

在 PHP 中，使用 array_push()函数可以向数组中添加元素，将传入的元素添加到某个数组的末尾，并返回数组新的单元总数。其语法如下：

```
int array_push ( array array, mixed var [, mixed ...])
```

参数 array 为指定的数组；参数 var 是压入数组中的值。

例 5.09　下面使用 array_push()函数向数组中添加元素，并输出添加元素后的数组，代码如下。**（实例位置：光盘\mr\05\sl\5.09）**

```
<?php
$array=array(0 =>'PHP 学习手册', 1 =>'JAVA 学习手册');  //声明数组
echo "添加前的数组元素：";
print_r($array);
echo "<br>";
array_push($array,'VB 学习手册','VC 学习手册');          //向数组中添加元素
echo "添加后的数组元素：";
print_r($array);                                         //输出添加后的数组结构
?>
```

运行结果如图 5.12 所示。

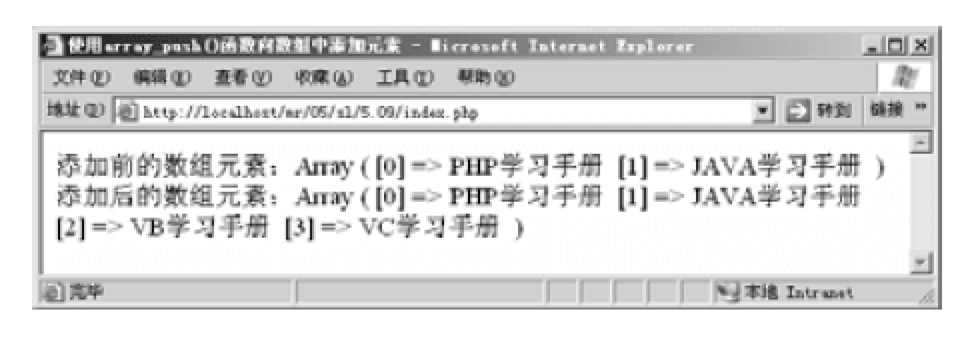


图 5.12　使用 array_push()函数向数组中添加元素

5.5.3　获取数组中最后一个元素

在 PHP 中，通过 array_pop()函数可以获取并返回数组中的最后一个元素，并将数组的长度减 1，如果数组为空（或者不是数组）将返回 null。其语法格式如下：

```
mixed array_pop ( array array)
```

参数 array 为输入的数组。

例 5.10　首先应用 array_push()函数向数组中添加元素，然后应用 array_pop()函数获取数组中最后一个元素，最后输出最后一个元素值，代码如下。**（实例位置：光盘\mr\05\sl\5.10）**

```
<?php
$array=array(0 =>'PHP24 堂课', 1 =>'JAVA24 堂课');        //声明数组
array_push($array,'VB24 堂课','VC24 堂课');               //向数组中添加元素
$last_array=array_pop($array);                            //获取数组中最后一个元素
echo $last_array;                                         //返回结果为 VC24 堂课
?>
```

运行结果为：VC24 堂课

5.5.4　删除数组中重复元素

在 PHP 中，使用 array_unique()函数可以将数组中重复的元素删除，语法如下：

```
array array_unique ( array array)
```

参数 array 为输入的数组。

说明：虽然 array_unique()函数只保留重复值的第一个键名。但是，这第一个键名并不是在未排序的数组中同一个值的第一个出现的键名，只有当两个字符串的表达式完全相同时 ((string) $elem1 === (string) $elem2)，第一个单元才被保留。

例 5.11 首先定义一个数组，然后应用 array_push()函数向数组中添加元素，并输出数组，最后应用 array_unique()函数删除数组中重复元素，并输出数组，代码如下。（**实例位置：光盘\mr\05\sl\5.11**）

```
<?php
$arr_int = array ("PHP", "JAVA","VC");          //定义数组
array_push ($arr_int, "PHP","VC");              //向数组中添加元素
print_r($arr_int);                              //输出添加后的数组
$result=array_unique($arr_int);                 //删除添加后数组中重复的元素
print_r($result);                               //输出删除重复元素后的数组
?>
```

运行结果如图 5.13 所示。

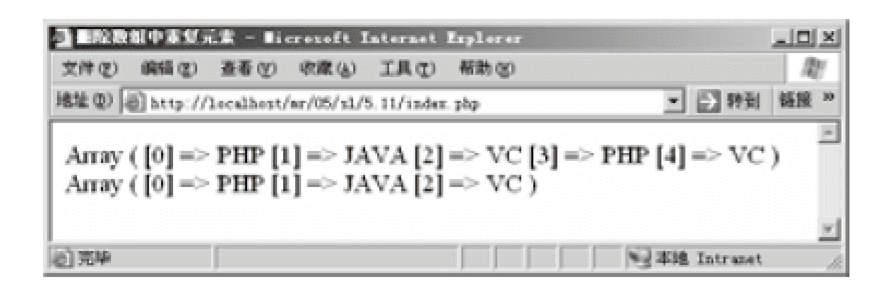


图 5.13 删除数组中重复元素

说明：使用 unset()函数可以删除数组中的某个元素，如将例 5.11 中的$arr_int 数组的第二个元素删除，关键代码如下：

```
unset($arr_int[1]);
```

5.5.5 获取数组中指定元素的键名

获取数组中指定元素的键名主要是通过数组函数来实现的，本小节将分别介绍使用 array_search()函数获取数组中指定元素的键名和使用 array_keys()函数获取数组中重复元素的所有键名的方法。

☑ 使用 array_search()函数获取数组中指定元素的键名

使用 array_search()函数获取数组中指定元素的键名，在数组中搜索给定的值，找到后返回键名，否则返回 FALSE。其语法如下：

```
mixed array_search ( mixed needle, array haystack [, bool strict])
```

array_search()函数的参数说明如表 5.1 所示。

表 5.1 array_search()函数的参数说明

参 数	说 明
needle	指定在数组中搜索的值，如果 needle 是字符串，则以区分大小写的方式进行比较
haystack	指定被搜索的数组
strict	可选参数，如果值为 TRUE，还将在 haystack 中检查 needle 的类型

说明：array_search()函数是区分字母大小写的。



例 5.12　下面使用 array_search()函数获取数组中元素的键名，具体代码如下。（**实例位置：光盘\mr\05\sl\5.12**）

```
<?php
$arr=array("苹果","桔子","香蕉","梨");        //创建数组，数组中有 4 个元素
$name=array_search("香蕉",$arr);
                    //使用 array_search()函数获取$arr 数组中“香蕉”的键名，然后将获取的结果赋给$name 变量
echo $name;                                    //输出结果
?>
```

运行结果为：2

☑　使用 array_keys()函数获取数组中重复元素的所有键名

如果查询的元素在数组中出现两次以上，那么 array_keys()函数则返回第一个匹配的键名。如果想要返回所有匹配的键名，则需要使用 array_keys()函数。其语法如下：

```
array array_keys ( array input [, mixed search_value [, bool strict]] )
```

array_keys()函数返回 input 数组中的数字或者字符串的键名。如果指定可选参数 search_value，则只返回该值的键名；否则 input 数组中的所有键名都会被返回。

例 5.13　下面使用 array_keys()函数来获取数组中重复元素的所有键名，具体代码如下。（**实例位置：光盘\mr\05\sl\5.13**）

```
<?php
$arr=array("苹果","桔子","香蕉","梨","香蕉");
$name=array_keys($arr,"香蕉");          //使用 array_keys()获取$arr 数组中“香蕉”的所有键值
print_r($name);                          //因为 array_keys()函数返回的是数组类型的值，所以使用 print_r 输出
?>
```

运行结果为：Array ([0] => 2 [1] => 4)

你问我答：如何完成字符串与数组的转换？

在程序开发过程中，字符串与数组的转换是必不可少的，并且非常实用。主要应用 explode()和 implode()函数实现，其中 explode()函数在第 3 堂课 3.3 节中已经进行了详细讲解，这里就不再赘述。

5.6　PHP 全局数组

应用 PHP 提供的全局数组，可以获取大量与环境有关的信息，如可以应用这些数组获取当前用户会话、用户操作环境和本地操作环境等信息。下面将对 PHP 中常用全局数组进行介绍。

5.6.1　$_SERVER[ ]全局数组

$_SERVER[ ]全局数组包含由 Web 服务器创建的信息，应用该数组可以获取服务器和客户配置及当前请求的有关信息。下面对$_SERVER[ ]数组进行介绍，如表 5.2 所示。

表 5.2　$_SERVER[ ]全局数组

数 组 元 素	说　　明
$_SERVER['SERVER_ADDR']	当前运行脚本所在的服务器的 IP 地址
$_SERVER['SERVER_NAME']	当前运行脚本所在服务器主机的名称。如果该脚本运行在一个虚拟主机上，该名称是由那个虚拟主机所设置的值决定

续表

数 组 元 素	说　　明
$_SERVER['REQUEST_METHOD']	访问页面时的请求方法，如 GET、HEAD、POST 和 PUT。如果请求的方式是 HEAD，PHP 脚本将在送出头信息后中止（这意味着在产生任何输出后，不再有输出缓冲）
$_SERVER['REMOTE_ADDR']	正在浏览当前页面用户的 IP 地址
$_SERVER['REMOTE_HOST']	正在浏览当前页面用户的主机名。反向域名解析基于该用户的 REMOTE_ADDR
$_SERVER['REMOTE_PORT']	用户连接到服务器时所使用的端口
$_SERVER['SCRIPT_FILENAME']	当前执行脚本的绝对路径名。注意：如果脚本在 CLI 中被执行，作为相对路径，如 file.php 或者../file.php，$_SERVER['SCRIPT_FILENAME']将包含用户指定的相对路径
$_SERVER['SERVER_PORT']	服务器所使用的端口。默认为 80。如果使用 SSL 安全连接，则这个值为用户设置 HTTP 端口
$_SERVER['SERVER_SIGNATURE']	包含服务器版本和虚拟主机名的字符串
$_SERVER['DOCUMENT_ROOT']	当前运行脚本所在的文档根目录。在服务器配置文件中定义

例 5.14　通过$_SERVER[ ]全局数组获取服务器和客户端的 IP 地址、客户端连接主机的端口号，以及服务器的根目录，代码如下。（**实例位置：光盘\mr\05\sl\5.14**）

```
<?php
    echo "当前服务器 IP 地址是：<b>".$_SERVER['SERVER_ADDR']."</b><br>";
    echo "当前服务器的主机名称是：<b>".$_SERVER['SERVER_NAME']."</b><br>";
    echo "客户端 IP 地址是：<b>".$_SERVER['REMOTE_ADDR']."</b><br>";
    echo "客户端连接到主机所使用的端口：<b>".$_SERVER['REMOTE_PORT']."</b><br>";
    echo "当前运行的脚本所在文档的根目录：<b>".$_SERVER['DOCUMENT_ROOT']."</b><br>";
?>
```

运行结果如图 5.14 所示。



图 5.14　获取服务器和客户端的 IP 地址

5.6.2　$_GET[ ]和$_POST[ ]全局数组

PHP 中提供$_GET[ ]和$_POST[ ]全局数组分别用来接收 GET 和 POST 方法传递到当前页面的数据。

例 5.15　下面开发一个实例，获取用户的登录信息。分别通过 GET 和 POST 方法完成数据的提交，并且应用$_GET[ ]和$_POST[ ]全局数组获取用户提交的数据，从返回的结果中体会两者之间的区别。（**实例位置：光盘\mr\05\sl\5.15**）

（1）创建 index.php 文件，同时定义两个 form 表单，分别使用 GET 和 POST 方法提交数据，将通过 GET 方法提交的数据传递到 get.php 文件，将通过 POST 方法提交的数据传递到 post.php 文件。

（2）创建 get.php 文件，通过$_GET[ ]全局数组获取 GET 方法提交的数据。运行结果如图 5.15 所示。





图 5.15　使用$_GET 变量输出页面

具体代码如下：

```
<?php
if(isset($_GET['Submit']) and $_GET['Submit']=="提交"){
	echo "用户名为：".$_GET['textfield']."<br>";
	echo "密码为：".$_GET['textfield2'];
}
?>
```

（3） 创建 post.php 文件，通过$_POST[ ]全局数组获取 POST 方法提交的数据。运行结果如图 5.16 所示。

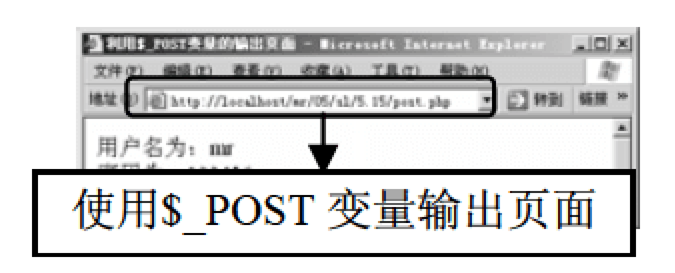


图 5.16　使用$_POST 变量输出页面

具体代码如下：

```
<?php
if(isset($_POST['Submit2']) and $_POST['Submit2']=="提交"){
	echo "用户名为：".$_POST['textfield3']."<br>";
	echo "密码为：".$_POST['textfield22'];
}
?>
```

5.6.3　$_COOKIE 全局数组

$_COOKIE[ ]全局数组存储通过 http Cookie 传递到脚本的信息。PHP 中可以通过 setCookie()函数设置 Cookie 的值，用$_COOKIE[ ]数组接收 Cookie 的值，$_COOKIE[ ]数组的下标为 Cookie 的名称。

例如，通过 setCookie()函数创建一个 Cookie，并通过$_COOKIE[ ]全局数组获取 COOKIE 的值。代码如下：

```
<?php
setCookie("mr",'明日科技');
setCookie("mr", '明日科技', time()+60);                //设置 Cookie 有效时间为 60 秒
echo "读取 Cookie：".$_COOKIE['mr'];                    //通过$_COOKIE[]读取 Cookie 的值
?>
```

5.6.4　$_ENV[ ]全局数组

$_ENV[ ]全局数组用于提供与服务器有关的信息。例如：

☑　$_ENV["HOSTNAME"]：获取服务器名称。

☑　$_ENV["SHELL"]：获取系统 shell。

5.6.5　$_REQUEST[ ]全局数组

可以用$_REQUEST[ ]全局数组获取 GET 方法、POST 方法和 http Cookie 传递到脚本的信息。如果在编写

程序时，不能确定是通过什么方法提交的数据，那么就可以通过$_REQUEST[ ]全局数组获取提交到当前页面的数据。

5.6.6 $_SESSION[ ]全局数组

$_SESSION[ ]全局数组用于获取会话变量的相关信息。

例如，初始化 SESSION 变量，通过$_SESSION[ ]全局数组为 SESSION 变量赋值，最后通过$_SESSION[ ]全局数组输出 SESSION 的值。代码如下：

```
<?php
session_start();                //初始化 SESSION 变量
$_SESSION['mr']="mrsoft";       //为 SESSION 变量赋值
echo   $_SESSION['mr']          //输出 SESSION 变量的值
?>
```

5.6.7 $_FILES[ ]全局数组

与其他全局数组不同，$_FILES[ ]全局数组为一个多维数组，该数组用于获取通过 POST 方法上传文件时的相关信息。如果为单文件上传，则该数组为二维数组；如果为多文件上传，则该数组为三维数组。下面对该数组的具体参数取值进行描述。

☑ $_FILES["file"]["name"]：从客户端上传的文件名称。
☑ $_FILES["file"]["type"]：从客户端上传的文件类型。
☑ $_FILES["userfile"]["size"]：已上传文件的大小。
☑ $_FILES["file"]["tmp_name"]：文件上传到服务器后，在服务器中的临时文件名。
☑ $_FILES["file"]["error"]：返回在上传过程中发生错误的错误代号。

注意：有关$_FILES[ ]全局数组的详细讲解请参考第 13 堂课。

5.7 照猫画虎——基本功训练

5.7.1 基本功训练 1——输出数组中元素值

视频讲解：光盘\mr\05\lx\输出数组中元素值.exe
实例位置：光盘\mr\05\zmhh\01

输出数组中的元素值有两种方法，一种是使用 print_r()函数，另一种是使用 echo 语句，其两种方法的区别在前面已经介绍过。下面使用 print_r()函数来输出$arr 数组中的元素值，运行结果如图 5.17 所示。

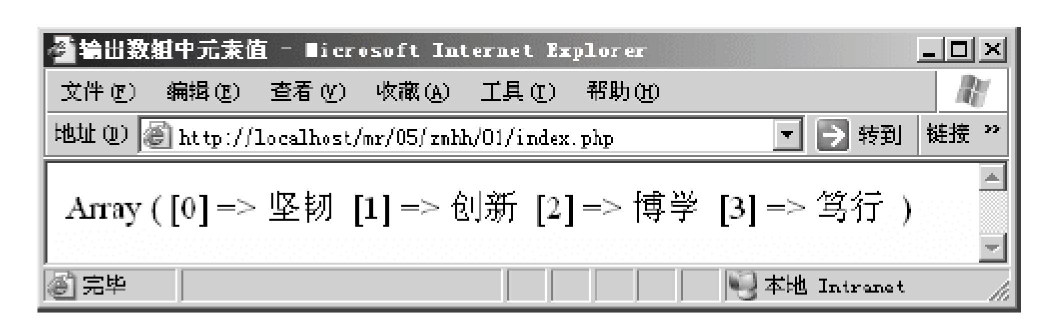
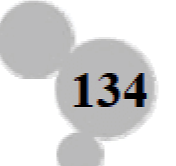


图 5.17 输出数组中的元素值

具体代码如下：



```
<?php
$arr=array("坚韧","创新","博学","笃行");
print_r($arr);
?>
```

照猫画虎：根据上面的例子，使用 echo 语句输出$arr 数组中的第二个和第 3 个元素值。提示：数组中元素的索引是从 0 开始计算的，所以数组中的第二个元素的索引应该是 1，依此类推。（10 分）（实例位置：光盘\mr\05\zmhh\01_zmhh）

5.7.2　基本功训练 2——遍历数组

视频讲解：光盘\mr\05\lx\遍历数组.exe

实例位置：光盘\mr\05\zmhh\02

前面介绍了 3 种遍历数组的方法，其中 list()和 each()函数只能遍历数字索引的数组。下面使用 list()和 each()函数来遍历数字索引数组$arr，运行结果如图 5.18 所示。



图 5.18　使用 list()和 each()函数遍历数组

具体代码如下：

```
<?php
$arr=array("坚韧","创新","博学","笃行");                 //创建数组
while(list($a,$b)=each($arr)){                          //使用 list()和 each()函数遍历数组
    echo $b;                                            //输出数组
}
?>
```

照猫画虎：使用 for 循环遍历上例中的数组$arr。（15 分）（实例位置：光盘\mr\05\zmhh\02_zmhh）

5.7.3　基本功训练 3——foreach 语句遍历数组

视频讲解：光盘\mr\05\lx\foreach 语句遍历数组.exe

实例位置：光盘\mr\05\zmhh\03

下面使用 foreach 语句遍历$arr 数组，并将数组元素的索引和值输出，运行结果如图 5.19 所示。



图 5.19　foreach 语句遍历数组

具体代码如下：

```
<?php
$arr=array("水果"=>"苹果","干果"=>"杏仁","零食"=>"薯片");              //创建数组
```

```
foreach($arr as $key=>$array){                                   //使用 foreach 语句遍历数组
      echo "$key-----$array"."<br>";                             //输出数组元素的索引和值
}
?>
```

照猫画虎：通过 foreach 语句遍历多维数组。提示：要遍历的数组是几维，就使用几个 foreach 语句。（15 分）（实例位置：光盘\mr\05\zmhh\03_zmhh）

5.7.4 基本功训练 4——将数组元素转换成字符串

视频讲解：光盘\mr\05\lx\将数组元素转换成字符串.exe

实例位置：光盘\mr\05\zmhh\04

将数组元素转换成字符串需要使用 implode()函数，下面使用该函数将数组$arr 中的元素值转换为字符串类型，运行结果如图 5.20 所示。



图 5.20 将数组元素转换成字符串运行结果

具体代码如下：

```
<?php
$arr=array("乒乓球","羽毛球","足球","网球");              //创建数组
$value=implode('*',$arr);   //使用 implode()函数将数组值转换为字符串类型，并以"*"隔开，再将值赋给变量 value
echo $value."<br>";         //输出变量 value 的值
echo is_string($value);     //使用 is_string()函数检验变量的值是否为字符串类型，并输出检测结果
?>
```

照猫画虎：接着上例，将转换后的字符串再转为数组类型。提示：将字符串转换为数组元素，需要使用 explode()函数，其使用方法和 implode()函数相同。转换为数组后，使用 print_r()函数进行输出。（10 分）（实例位置：光盘\mr\05\zmhh\04_zmhh）

5.7.5 基本功训练 5——去除数组中重复的元素

视频讲解：光盘\mr\05\lx\去除数组中重复的元素.exe

实例位置：光盘\mr\05\zmhh\05

去除数组中重复的元素可以使用 array_unique()函数，下面去除$arr 数组中的重复元素，运行结果如图 5.21 所示。



图 5.21 去除数组中重复的元素



具体代码如下：

```
<?php
$arr=array("水果"=>"苹果","干果"=>"杏仁","零食"=>"杏仁");        //创建数组，数组中有两个元素的值是相同的
$value=array_unique($arr);                                    //使用 array_unique 函数去除数组中的重复记录
print_r($value);                                              //输出删除重复记录后的数组元素
?>
```

照猫画虎：删除上例中值为“苹果”的元素，然后输出删除后的数组。提示：使用 unset()函数可以删除数组中指定元素。（15 分）（实例位置：光盘\mr\05\zmhh\05_zmhh）

5.7.6　基本功训练 6——向数组中添加元素

视频讲解：光盘\mr\05\lx\向数组中添加元素.exe

实例位置：光盘\mr\05\zmhh\06

使用 array_push()函数可实现向数组中添加元素，下面使用该函数向$arr 数组中添加两个元素，运行结果如图 5.22 所示。



图 5.22　向数组中添加元素

具体代码如下：

```
<?php
$arr=array("坚韧","创新");              //创建数组
echo "原数组元素：";
print_r($arr);
echo "<br>";
array_push($arr,"博学","笃行");         //向数组中添加两个元素
echo "添加后的数组元素：";
print_r($arr);                          //输出添加元素后的数组
?>
```

照猫画虎：向上例中的$arr 数组中添加一个元素为“把握机遇 用业绩体现价值”，然后输出数组。（20 分）（实例位置：光盘\mr\05\zmhh\06_zmhh）

5.7.7　基本功训练 7——获取服务器端的 IP 地址

视频讲解：光盘\mr\05\lx\获取服务器端的 IP 地址.exe

实例位置：光盘\mr\05\zmhh\07

获取服务器端的 IP 地址使用$_SERVER[]全局数组中的$_SERVER['SERVER_ADDR']，下面使用该数组元素获取服务器端的 IP 地址，运行结果如图 5.23 所示。



图 5.23　获取服务器端的 IP 地址

具体代码如下：

```
<?php
echo "服务器端的 IP 地址为：".$_SERVER['SERVER_ADDR'];
?>
```

照猫画虎：获取客户端 IP 地址。提示：使用$_SERVER[]全局数组中的$_SERVER['REMOTE_ADDR']元素可以实现该功能。（15 分）（实例位置：光盘\mr\05\zmhh\07_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	6	7	总分数	
分数									

5.8　情景应用——拓展与实践

5.8.1　情景应用 1——字符串数组排序

视频讲解：光盘\mr\05\lx\字符串数组排序.exe

实例位置：光盘\mr\05\qjyy\01

PHP 中提供了很多对数组进行排序的函数，本例中使用较为常用的 sort()函数来对数组进行排序。sort()函数用于实现对数组元素从低到高排序，对于数值型的数组则按数值的大小进行排序；字符串型数组则按 ASCII 码的顺序排序。sort()函数的语法如下：

```
bool sort ( array &array [, int sort_flags] )
```

参数 array 指定要进行排序的数组；参数 sort_flags 指定排序的方式，该参数的可取值如表 5.3 所示。

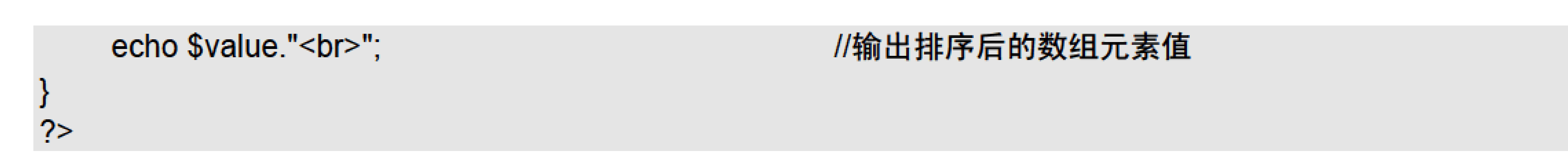
表 5.3　sort_flags 参数的可取值

参　数　值	意　　义
SORT_REGULAR	默认值，正常比较数组元素
SORT_NUMERIC	将数组元素作为数字来比较
SORT_STRING	将数组元素作为字符串来比较

下面使用 sort()函数对数组$arr 中的元素值进行排序，运行结果如图 5.24 所示。

具体代码如下：

```
<?php
$arr=array("足球","排球","网球","100 米短跑");                //创建数组
sort($arr);                                                   //对数组进行排序
foreach($arr as $value){                                      //使用 foreach 语句循环输出排序后的数组
```



```
    echo $value."<br>";                              //输出排序后的数组元素值
}
?>
```



图 5.24　使用 sort()函数对数组进行排序

DIY：使用 rsort()函数对上例中的$arr 数组元素进行排序。提示：rsort()函数用于实现对数组元素进行从高到低的排序，用法和 sort()函数相同。（15 分）（实例位置：光盘\mr\05\qjyy\01_diy）

5.8.2　情景应用 2——获取用户注册信息

视频讲解：光盘\mr\05\lx\获取用户注册信息.exe

实例位置：光盘\mr\05\qjyy\02

用户注册是大多数网站都具备的功能，本例中设计的用户注册信息包括用户名、密码、性别、学历、爱好、个人简介，信息填写完成后，单击“提交”按钮，将信息输出到页面中。运行效果如图 5.25 所示。

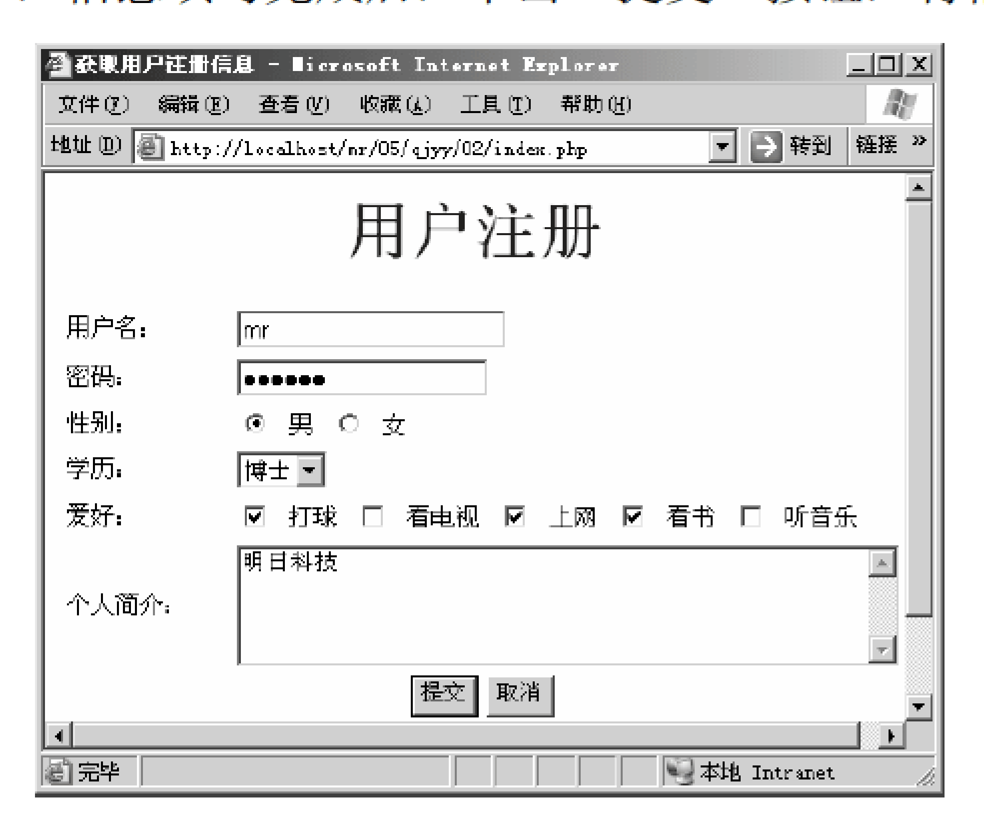




图 5.25　获取用户注册信息

从运行结果可以看出用户名、密码、性别、学历、个人简介都是唯一的，只有爱好可以是多个，则说明爱好为数组类型的值。获取用户注册信息的具体代码如下：

```
<?php
echo "用户名：".$_POST['textfield']."<br>";             //输出用户名
echo "密码：".$_POST['textfield2']."<br>";              //输出密码
echo "性别：".$_POST['radiobutton']."<br>";             //输出性别
echo "学历：".$_POST['select']."<br>";                  //输出学历
//下面为循环输出爱好信息
echo "爱好：";
$array=$_POST['checkbox'];
foreach($array as $arr){
    echo $arr."  ";
```

```
}
echo "<br>";
echo "个人简介：".$_POST['textarea'];                        //输出个人简介
?>
```

本例主要将数组的应用融入到了实际编程中，也应用了表单方面的知识，此处只要了解数组方面的内容即可，至于表单相关内容将在第 6 堂课中详细介绍。

DIY：对上例稍做改动，制作一个简单的用户登录界面，并获取用户的登录信息。提示：结合上例中的输出用户名和密码的方法。（20 分）（实例位置：光盘\mr\05\qjyy\02_diy）

5.8.3　情景应用 3——获取上传文件的数据

视频讲解：光盘\mr\05\lx\获取上传文件的数据.exe

实例位置：光盘\mr\05\qjyy\03

获取上传文件的数据，需要使用$_FILES[]全局数组中的$_FILES["file"]["name"]实现，下面使用该全局数组获取上传文件的名称，运行结果如图 5.26 所示。



图 5.26　获取上传文件的数据

具体代码如下：

```
<?php
if(isset($_POST['Submit']) and $_POST['Submit']=="获取上传文件名"){     //判断按钮的值是否存在，是否为空
      echo $_FILES["file"]["name"];                                    //输出上传文件名
}
?>
```

DIY：结合上面的例子，获取上传文件的大小并输出。提示：使用$_FILES 全局数组中的$_FILES ["userfile"]["size"]，其中["userfile"]为文件域 name 属性的值，具体可参见上面的例子。（20 分）（实例位置：光盘\mr\05\qjyy\03_diy）

5.8.4　情景应用 4——生成在线考试题

视频讲解：光盘\mr\05\lx\生成在线考试题.exe

实例位置：光盘\mr\05\qjyy\04

本例实现在线生成考试题，将考题以字符串形式上传，然后使用 explode()函数将字符串转换为数组，并以*进行分隔，然后使用 for 循环输出数组元素值。运行结果如图 5.27 所示。





图 5.27　生成在线考试题

具体代码如下：

```
<?php
if(isset($_POST['Submit']) && $_POST['Submit']!=""){
	$content=$_POST['content'];		//获取表单提交的数据
	$data=explode("*",$content);		//应用 explode()函数对表单提交的数据进行拆分，以*号为分隔符
	for($i=0;$i<count($data);$i++){		//for 循环输出数组中的数据
		echo $data[$i]."<br><br>";
	}
}
?>
```

DIY：参照上面的例子，实现添加多选题功能。提示：添加多选题以*分隔，然后使用 explode()函数将字符串转换为数组。循环输出时注意复选框的添加。（20 分）（实例位置：光盘\mr\05\qjyy\04_diy）

5.8.5　情景应用 5——通过客户端 IP 地址限制投票次数

视频讲解：光盘\mr\05\lx\通过客户端 IP 地址限制投票次数.exe

实例位置：光盘\mr\05\qjyy\05

下面制作一个简单的投票系统，通过获取客户端 IP 地址来限制用户的投票次数，每个 IP 只可以投票一次，如果重复投票将给出提示。运行结果如图 5.28 所示。



图 5.28　在线投票

获取客户端 IP 地址需要使用$_SERVER[]全局数组中的$_SERVER['REMOTE_ADDR']参数。本例将客户端 IP 地址存放到数据库中，当用户提交投票时，使用 if…else 语句判断该 IP 在数据库中是否存在，从而实现通过客户端 IP 地址限制投票次数，具体代码如下：

```
<?php
//连接数据库
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database05",$conn);
mysql_query("set names utf8");
$ip=$_SERVER['REMOTE_ADDR'];                    //获取客户端 IP 地址
$insert="insert into tb_vote(IP)values('$ip')";          //定义添加数据语句
$select="select * from tb_vote where ip='$ip'";         //定义查询语句，查询 IP 地址在数据库中是否存在
if(isset($_POST['Submit']) and $_POST['Submit']=="提交投票"){     //判断按钮的执行操作
    $value=mysql_query($select,$conn);              //执行查询操作
    if(mysql_num_rows($value)==0){                  //判断查询结果是否等于 0
        $result=mysql_query($insert,$conn);         //如果等于 0，执行添加语句，将 IP 地址添加到数据库中
        if($result){                                //判断添加操作是否成功执行
            echo "<script>alert('投票成功！');window.location.href='index.php';</script>";
                                                    //如果成功执行，则提示投票成功
        }else{
            echo "<script>alert('投票失败！');window.location.href='index.php';</script>";
                                                    //否则提示投票失败
        }
    }else{
        echo "<script>alert('您已经投过票了！');window.location.href='index.php';</script>";
    }
}
?>
```

DIY：根据上面的例子，限制每个 IP 可以投 3 次票。（25 分）（实例位置：光盘\mr\05\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

5.9 自我测试

一、选择题（每题 10 分，5 道题）

1．统计数组元素个数的函数是（　　）。

A．array()　　B．count()　　C．foreach　　D．list()

2．创建数组的函数是（　　）。

A．array()　　B．count()　　C．foreach　　D．list()

3．输出数组时，echo()、print()和 print_r()的区别是（　　）。

A．echo()和 print_r()只可以输出数组中的某个元素值

B．print_r()可以输出数组，而 echo()和 print()只可以输出数组中的某个元素值

C．echo()只可以输出数组中的某个元素值，print()和 print_r()可以输出数组

D．没什么区别

4．PHP 中常用的遍历数组方法有几种？分别是什么？（　　）

A．1 种，即 foreach 循环语句

B．3 种，即 foreach 循环语句、for 循环语句和 list()函数

C．3 种，即 foreach 循环语句、for 循环语句、list()函数和 each()函数

D．4 种，即 foreach 循环语句、for 循环语句、list()函数和 each()函数、while 循环语句

5．删除数组中的某个元素，使用（　　）函数。

A．count()　　B．array()　　C．list()　　D．unset()

二、填空题（每题 10 分，5 道题）

1．下面是获取已上传文件的大小，请将代码补充完整。

```
<?php
echo $_FILES["userfile"]["_______"];
?>
```

2．下面向数组$array 中添加名元素，请将代码补充完整。

```
<?php
$array=array("长春","吉林","黑龙江");
_______($array,'上海');
?>
```

3．删除数组中的元素，使用（　　）函数。

4．在 PHP 中将以数字作为键名的数组称为（　　）。

5．根据数组维数的不同，可以把数组分为（　　）、（　　）和（　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

5.10　行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　容	行 动 指 南	
1	照猫画虎栏目 分数（　　）	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
		分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目 分数（　　）	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
		分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。

续表

序号	内　　容	行 动 指 南	
1	自我测试栏目 分数（　　）	分数>75 分	优秀，有成为编程高手的潜质。
		分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）创建一个数组，然后将数组元素打印出来。	
		（2）创建一个数组，然后使用 foreach 循环语句遍历数组元素。	
		（3）创建一个数组，获取数组元素个数，并输出获取到的数组元素个数。	
		（4）创建一个具有重复元素的数组，然后将重复的元素删除。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

5.11　成功可以复制——迅雷创始人邹胜龙

成就：2002 年底创建迅雷公司，2003 年回国发展，如今，迅雷的市场覆盖率达 80%。

创始公司：迅雷网络技术有限公司

财富巅峰：资产范围：9 亿以上

出生日期：1972 年

最高学历：计算机科学硕士

超级语录：迅雷的团队是一支学习能力和执行能力都很强的队伍，他们是中国互联网最优秀的团队之一。

成功历程

“迅雷”于 2002 年底由邹胜龙及程浩先生始创于美国硅谷。2003 年 1 月底，邹胜龙回国发展，在深圳市创办三代科技开发有限公司。2005 年 5 月，公司正式更名为深圳市迅雷网络技术有限公司，就是下载速度可达到迅雷不及掩耳之势。

“迅雷”立足于为全球互联网提供最好的多媒体下载服务。经过艰苦创业——“迅雷”在大中华区域



以领先的技术和诚信的服务，赢得了广大用户的深深喜爱和许多合作伙伴的认同与支持。公司旗舰产品——迅雷，已经成为中国互联网最流行的应用服务软件之一。作为中国最大的下载服务提供商，迅雷每天服务来自几十个国家，超过数千万次的下载。现在迅雷每天为全球互联网传送 3200000GB 的资源，并以其下载软件的迅捷高速，每天为人类节省 1500 年时间。在本土，迅雷的市场覆盖率达 85%，迄今，全球已有 1.88 亿网民体验过了迅雷提供的服务。随着中国互联网宽带的普及，迅雷凭借“简单、高速”的下载体验，已经成为高速下载的代名词。此外，“迅雷”也获得了晨光科技和 IDGVC 等数家知名风险投资企业的认同和合资。2007 年 1 月迅雷宣布第三次融资成功，本次融资的领衔投资是联创策源（Ceyuan Ventures），参与投资的有晨光（Morningside Ventures）、IDGVC 和 Fidelity Asia Ventures，战略投资是 Google（谷歌）。这些投资合作伙伴除了给“迅雷”带来了更加雄厚的资金实力，也给“迅雷”带来了更丰富的行业资源和国际化公司运行环境。



迅雷界面



在网易科技采访邹胜龙时说“您已经入选超过 5 万网友选出的互联网领袖扑克牌，并且进入了前十名，方便谈一下您的感受吗？”邹胜龙说：“非常荣幸！不过，我个人感觉受之有愧！其实，大家对我的厚爱主要是来自于对于迅雷的认识，而迅雷主要是由我的合伙人程浩和我们的团队一起建设的。相比之下，迅雷还非常年轻，还有很长的道路要走下去。”通过邹先生的这一番话，就可以了解到，任何一个成功的项目都不会是一个人完成的，想要在 IT 领域中获得成功，就要做好团队合作的准备，凝聚大家的力量，发挥大家的聪明才智，才能创造财富。

第 6 堂课

PHP 与 Web 页面交互

（视频讲解：225 分钟）

表单是 Web 编程中不可缺少的重要元素，一般作为收集用户提交的数据。表单程序的运作原理是：在客户端通过表单提交数据，将数据提交给 Web 服务器的脚本程序。在脚本程序中完成对数据的处理。下面讲解表单以及表单中的元素是如何定义的。

学习摘要：

- **创建表单**
- **获取表单提交的数据**
- **获取超链接传递的数据**
- **了解 JavaScript 脚本**

6.1 表单概述

Web 表单的功能是让浏览者和网站有一个互动的平台，主要用来在网页中发送数据到服务器。例如，提交注册信息时，当用户填写完信息后执行提交操作，将表单中的数据从客户端的浏览器传送到服务器端，经过服务器端 PHP 程序进行处理后，再将用户所需要的信息传递回客户端的浏览器上，从而获得用户信息，使 PHP 与 Web 表单实现交互。

6.1.1 创建表单

表单是使用<form></form>标签来创建并定义表单的开始和结束位置，中间包含多个元素，表单结构如下：

```
<form name="form_name"  method="method" action="url"  enctype="value"  target="target_win"  id="id">
……                                  //省略插入的表单元素
</form >
```

<form>标记的属性如表 6.1 所示。

表 6.1　<form>标记的常用属性

属　性	描　述
name	表单名称
id	表单的 ID 号
method	该属性用于定义表单中数据的提交方式，可取值为 GET 和 POST 中的一个。GET 方法将表单内容附加在 URL 地址后面进行提交，所以对提交信息的长度进行了限制，不可以超过 8192 个字符，同时 GET 方法不具有保密性，不适合处理如信用卡卡号等要求保密的内容，而且不能传送非 ASCII 的字符；POST 方法将用户在表单中填写的数据包含在表单的主体中，一起传送到服务器，不会在浏览器的地址栏中显示，这种方式传送的数据没有大小限制。默认为 GET 方法
action	该属性定义将表单中的数据提交到哪个文件中进行处理，这个地址可以是绝对 URL，也可以是相对的 URL。如果这个属性是空值则提交到当前文件
enctype	设置表单资料的编码格式
target	该属性和链接中的同名属性类似，用来指定目标窗口或目标帧

6.1.2 添加表单元素

表单（form）由表单元素组成。常用的表单元素有以下几种标记：输入域标记<input>、选择域标记<select>和<option>、文本域标记<textarea>等，下面分别进行介绍。

1．输入域标记<input>

输入域标记<input>是表单中最常用的标记之一。常用的文本域、按钮、单选按钮、复选框等构成了一个完整的表单。其语法格式如下：

```
<form>
<input name="filed_name"  type="type_name">
</form>
```

其中参数 name 是指输入域的名称，参数 type 是指输入域的类型。在<input type="">标记中一共提供了 10 种类型的输入区域，用户所选择使用的类型由 type 属性决定。type 属性取值及举例如表 6.2 所示。

表 6.2　type 属性取值及举例

值	举　例	说　明	运 行 结 果
text	<input name="user" **type="text"** value="纯净水" size="12" maxlength="1000">	name 为文本框的名称，value 是文本框的默认值，size 指文本框的宽度（以字符为单位），maxlength 指文本框的最大输入字符数	添加一个文本框： 纯净水
hidden	<input **type="hidden"** name= "ddh">	隐藏域，用于在表单中以隐含方式提交变量值。隐藏域在页面中对于用户而言是不可见的，添加隐藏域的目的在于通过隐藏的方式收集或者发送信息。浏览者单击发送按钮发送表单时，隐藏域的信息也被一起发送到 action 指定的处理页	添加一个隐藏域：
password	<input name="pwd" **type="password"** value="666666" size="12" maxlength="20">	密码域，用户在该文本框中输入字符时将被替换显示为*号，起到保密作用	添加一个密码域： ******
file	<input name="file" **type="file" enctype="multipart/form-data"** size="16" maxlength="200">	文件域，当文件上传时，可用来打开一个模式窗口以选择文件。然后将文件通过表单上传到服务器，如上传 Word 文件等各种类型的文件。但是必须注意的是，上传文件时需要指明表单的属性 enctype= "multipart/form-data"才可以实现上传功能	添加一个文件域： 浏览...
image	<input name="imageField" **type="image"** src="images/banner.gif" width="120" height="24" border="0">	图像域是指可以用在提交按钮位置上的图片，这幅图片具有按钮的功能	添加一个图像域：
radio	<input name="sex" **type= "radio"** value="1" **checked**>男 <input name="sex" **type= "radio"** value="0">女	单选按钮，用于设置一组选择项，用户只能选择一项 checked 属性用来设置单选按钮默认值	添加一组单选按钮（如您的性别为：） ◉ 男 ○ 女
checkbox	<input name="checkbox" **type= "checkbox"** value="1" **checked**>封面 <input name="checkbox" **type= "checkbox"** value="1" **checked**>正文内容 <input name="checkbox" **type= "checkbox"** value="0">价格	复选框，允许用户选择多个选择项。checked 属性用来设置该复选框默认值。例如，收集个人信息时，要求在个人爱好的选项中进行多项选择等	添加一组复选框（如影响您购买本书的因素：） ☑ 封面 ☑ 正文内容 ☐ 价 格
submit	<input **type="submit"** name= "Submit" value="提交">	将表单的内容提交到服务器端	添加一个提交按钮： 提交
reset	<input **type="reset"** name= "Submit" value="重置">	清除与重置表单内容，用于清除表单中所有文本框的内容，而且使选择菜单项恢复到初始值	添加一个重置按钮： 重置
button	<input **type="button"** name= "Submit" value="按钮">	按钮可以激发提交表单的动作，可以在用户需要修改表单时，将表单恢复到初始的状态，还可以依照程序的需要，发挥其他作用。普通按钮一般是配合 JavaScript 脚本来进行表单的处理	添加一个普通按钮： 按钮

2. 选择域标记<select>和<option>

通过选择域标记<select>和<option>可以建立一个列表或者菜单。菜单节省空间，正常状态下只能看到一个选项，单击按钮打开菜单后才能看到全部的选项。列表可以显示一定数量的选项，如果超出了这个数量，会自动出现滚动条，浏览者可以通过拖动滚动条来查看各选项。其语法格式如下：

```
<select name="name" size="value" multiple>
<option value="value" selected>选项 1</option>
<option value="value">选项 2</option>
<option value="value">选项 3</option>
…
</select>
```

其中参数 name 表示选择域的名称；参数 size 表示列表的行数；参数 value 表示菜单选项值；参数 multiple 表示以菜单方式显示数据，省略则以列表方式显示数据。

选择域标记<select>和<option>的显示方式及举例如表 6.3 所示。

表 6.3　选择域标记<select>和<option>的显示方式及举例

显示方式	举　例	说　明	运行结果
列表方式	<select name="spec" id="spec"> <option value="0" selected>网络编程</option> <option value="1">办公自动化</option> <option value="2">网页设计</option> <option value="3">网页美工</option> </select>	下拉列表框，通过选择域标记<select>和<option>建立一个列表，列表可以显示一定数量的选项，如果超出了这个数量，会自动出现滚动条，浏览者可以通过拖动滚动条来查看各选项。selected 属性用来设置该菜单项时默认被选中	请选择所学专业： 网络编程 网络编程 办公自动化 网页设计 网页美工
菜单方式	<select name="spec" id="spec" multiple > <option value="0" selected>网络编程</option> <option value="1">办公自动化</option> <option value="2">网页设计</option> <option value="3">网页美工</option> </select>	multiple 属性用于下拉列表<select>标记中，指定该选项用户可以使用 Shift 和 Ctrl 键进行多选	请选择所学专业： 网络编程 办公自动化 网页设计 网页美工

> 说明：表 6.3 中给出了静态菜单项的添加方法，而在 Web 程序开发过程中，也可以通过循环语句动态添加菜单项。

3. 文本域标记<textarea>

文本域标记<textarea>用来制作多行的文本域，可以在其中输入更多的文本。其语法格式如下：

```
<textarea name="name" rows=value cols=value value="value" warp="value">
    …文本内容
</textarea>
```

其中，参数 name 表示文本域的名称，rows 表示文本域的行数，cols 表示文本域的列数（这里的 rows 和 cols 以字符为单位），value 表示文本域的默认值。warp 用于设定显示和送出时的换行方式，值为 off 表示不自动换行；值为 hard 表示自动按 Enter 键换行，换行标记一同被发送到服务器，输出时也会换行；值为

soft 表示自动按 Enter 键换行，换行标记不会被发送到服务器，输出时仍然为一列。

文本域标记<textarea>的值及举例如表 6.4 所示。

表 6.4　文本域标记<textarea>的值及举例

值	举　　例	说　　明	运 行 结 果
textarea	<**textarea** name="remark" cols="20" rows="4" id="remark"> 请输入您的建议! </**textarea**>	文本域，也称多行文本框，用于多行文本的编辑。默认 warp 属性时，默认为自动换行方式	请发表您的建议： 请输入您的建议！

例 6.01　下面创建一个表单，表单元素包含文本域、单选按钮、复选框、下拉列表和提交按钮等，具体代码如下。（**实例位置：光盘\mr\06\sl\6.01**）

```
<form id="form1" name="form1" method="post" action="">
  <table width="286" border="0" align="center">
    <tr>
      <td width="72"><span class="STYLE1">用户名：</span></td>
      <td width="204"><label>
        <input type="text" name="textfield" />
      </label></td>
    </tr>
    <tr>
      <td><span class="STYLE1">密码：</span></td>
      <td><label>
        <input type="password" name="textfield2" />
      </label></td>
    </tr>
    <tr>
      <td><span class="STYLE1">性别：</span></td>
      <td><label>
        <input name="radiobutton" type="radio" value="radiobutton" checked="checked" />
        <span class="STYLE1">男          </span>
        <input type="radio" name="radiobutton" value="radiobutton" />
      <span class="STYLE1">女</span></label></td>
    </tr>
    <tr>
      <td><span class="STYLE1">学历：</span></td>
      <td><label>
        <select name="select">
          <option>高中</option>
          <option>大专</option>
          <option>本科</option>
          <option>博士</option>
        </select>
      </label></td>
    </tr>
    <tr>
      <td><span class="STYLE1">爱好</span></td>
      <td><label>
```

```
            <input type="checkbox" name="checkbox" value="checkbox" />
            <span class="STYLE1">打球</span>
        <input type="checkbox" name="checkbox2" value="checkbox" />
        <span class="STYLE1">看书</span>
        <input type="checkbox" name="checkbox3" value="checkbox" />
        <span class="STYLE1">听音乐</span></label></td>
    </tr>
    <tr>
        <td><span class="STYLE1">个人简介：</span></td>
        <td><label>
            <textarea name="textarea" rows="5"></textarea>
        </label></td>
    </tr>
    <tr>
        <td colspan="2" align="center"><label>
            <input type="submit" name="Submit" value="提交" />
        </label>
            <label>
            <input type="submit" name="Submit2" value="取消" />
            </label></td>
    </tr>
  </table>
  <p>
    <label></label>
  </p>
</form>
```

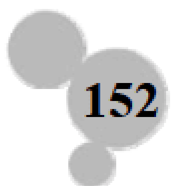
上述表单创建的界面效果如图 6.1 所示。



图 6.1　表单创建的界面效果

6.1.3　定义表单数据提交方法

表单数据的提交方法有两种，即 POST 方法和 GET 方法。设置表单提交数据的方法非常简单，只需设置<form>表单中的 method 属性值即可，如下所示：



```
<form id="form1" name="form1" method="POST" action="">
```

或

```
<form id="form1" name="form1" method="GET" action="">
```

这两种方法在 Web 页面的应用上有着本质的不同，下面分别对这两种方法进行介绍。

☑　POST 方法

POST 方法不依赖于 URL，不会将传递的参数值显示在地址栏中。另外，POST 方法可以没有限制地传递数据到服务器，所有提交的信息在后台传输，用户在浏览器端是看不到这一过程的，安全性高。所以，POST 方法比较适合用于发送一个保密的（如信用卡号）或者大量的数据到服务器。

例 6.02　下面使用 POST 方法提交表单信息到服务器，HTML 代码如下。（**实例位置：光盘\mr\06\sl\6.02**）

```
<form id="form1" name="form1" method="POST" action="">
  卡号：
  <label>
  <input type="text" name="textfield" />
  </label>
  <label>
  <input type="submit" name="Submit" value="提交" />
  </label>
</form>
```

执行程序，输入卡号，单击“提交”按钮，运行结果如图 6.2 所示。

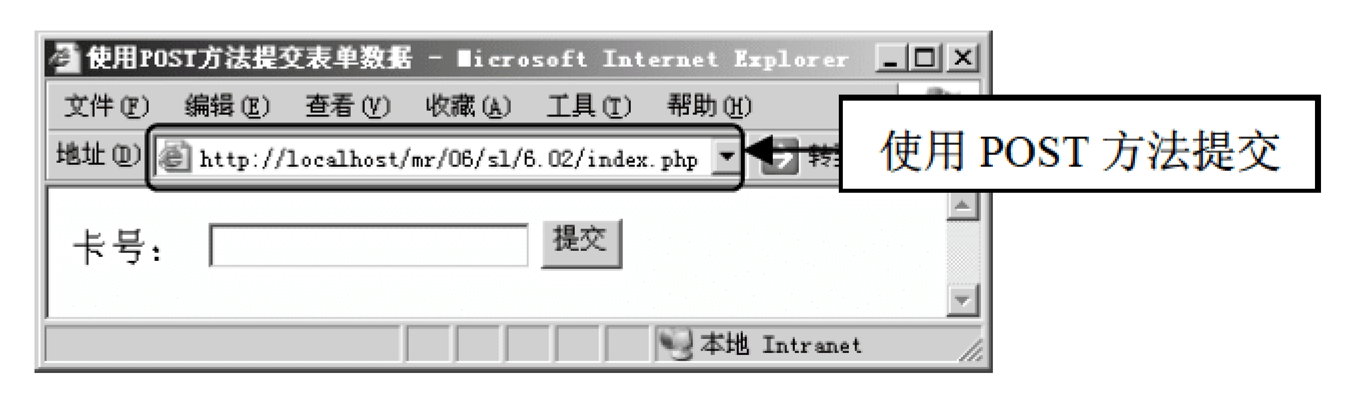


图 6.2　使用 POST 方法提交表单数据

☑　GET 方法

GET 方法是<form>表单中 method 属性的默认方法。使用 GET 方法提交的表单数据被附加到 URL 上，并作为 URL 的一部分发送到服务器端。在程序的开发过程中，由于 GET 方法提交的数据是附加到 URL 上发送的，因此，在 URL 的地址栏中将会显示“URL+用户传递的参数”。

GET 方法的传参格式如下：

```
http://url?name1=value1&name2=value2……
```

URL　　参数 1　　参数 2，也称查询字符

其中，url 为表单响应地址（如 127.0.0.1/index.php），name1 为表单元素的名称，value1 为表单元素的值。url 和表单元素之间用“?”隔开，而多个表单元素之间用“&”隔开，每个表单元素的格式都是“name=value”固定不变。

注意： 若要使用 GET 方法发送表单，URL 的长度应限制在 1MB 字符以内。如果发送的数据量太大，数据将被截断，从而导致意外或失败的处理结果。

例 6.03　将例 6.02 中的 method 属性值改为 GET，执行程序，运行结果如图 6.3 所示。（**实例位置：光盘\mr\06\sl\6.03**）

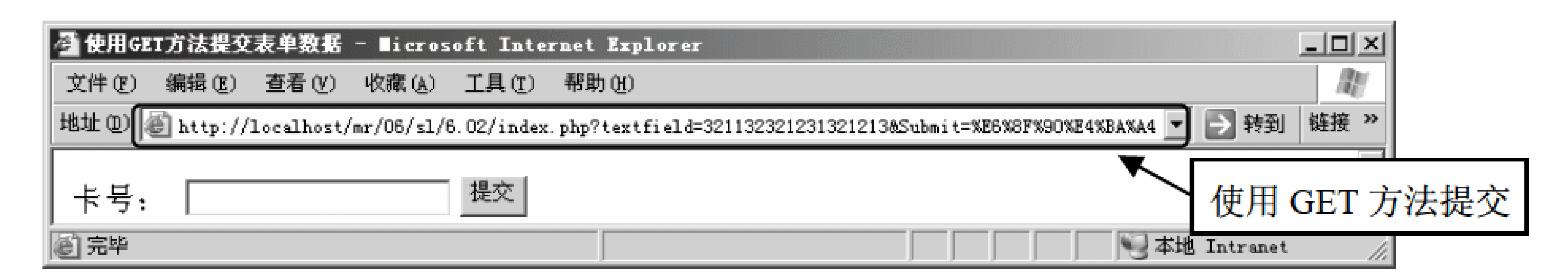


图 6.3　使用 GET 方法提交表单数据

6.2　获取表单提交的数据

获取表单提交数据有两种方法，分别为$_GET[]和$_POST[]，在实际程序开发过程中，使用哪种方法获取数据资源，是由<form>表单的 method 属性决定的。如果表单中 method 属性指定的是用 POST 方法进行数据传递，那么在处理数据时就应该使用$_POST[]全局变量获取表单数据。

6.2.1　$_GET[]获取表单提交数据

使用$_GET[]全局变量获取表单元素的格式如下：

```
$_GET['name']
```

例如，创建一个表单，设置 method 属性值为 GET，在表单中添加一个文本框，设置 name 属性为 user。获取表单元素值的代码如下：

```
<?php
$user=$_GET["user"];                    //应用$_GET[]全局变量获取表单元素中文本框的值
?>
```

6.2.2　$_POST[]获取表单提交数据

使用$_POST[]全局变量获取表单元素的格式如下：

```
$_POST[name]
```

例如，建立一个表单，设置 method 属性值为 POST，在表单中添加一文本框，设置 name 属性为 user。获取表单元素值的代码如下：

```
<?php
$user=$_POST["user"];                   //应用$_POST[]全局变量获取表单元素中文本框的值
?>
```

注意： PHP 可以应用$_POST[]或$_GET[]全局变量来获取表单元素的值。但值得注意的是，获取的表单元素名称区分字母大小写。如果在编写 Web 程序时忽略字母大小写，那么在程序运行时将获取不到表单元素的值或弹出错误提示信息。

<form>标签中所有表单元素都具备 name 属性，即为表单元素的名称，在使用时需要使用 name 属性来获取相应的 value 属性值。所以添加的所有控件必须定义其 name 属性值，另外，为了避免获取的数据出现错误，表单元素在命名上尽可能不要重复，尽量使用具有一定意义的英文缩写或拼音命名。

在程序开发过程中，获取文本框、密码域、隐藏域、按钮以及文本域的值的方法是相同的，都是一个表单元素对应一个值，通过 name 属性来获取相应的 value 属性值即可。

例 6.04　下面以获取文本框中的数据信息为例，讲解获取表单数据的方法。希望读者能够举一反三，



自行完成获取其他值。**（实例位置：光盘\mr\06\sl\6.04）**

具体操作步骤如下：

（1）应用开发工具（如 Dreamweaver），新建一个 PHP 动态页，存储为 index.php，如图 6.4 所示。



图 6.4　应用 Dreamweaver 开发工具创建 PHP 动态页

（2）在<form>表单元素外的任意位置添加 PHP 标记符，应用 if 条件语句判断用户是否提交了表单，如果条件成立，则通过$_POST[]全局变量获取用户提交的文本框“$_POST['accounts']”中的值，代码如下：

```
<?php
if(isset($_POST['Submit']) and $_POST['Submit']=="提交"){
                                        //if 语句判断 name 值为 Submit 的按钮是否存在，并且是否等于提交
    echo "您输入的卡号为：".$_POST['accounts'];
                                        //如果上面的条件成立，则输出 name 值为 accounts 的文本框中的值
}
?>
```

（3）在 IE 浏览器中输入地址，按 Enter 键，输入卡号，单击“提交”按钮即可，运行结果如图 6.5 所示。



图 6.5　获取文本框的信息

注意： 在应用文本框传递值时，一定要正确书写文本框的名称，在表单元素的命名上不应该有空格存在；在获取文本框提交的值时，书写的文本框名称一定要与提交文本框页中设置的名称相同，否则将不能获取到文本框的值。

6.3　获取超链接传递的数据

通过超链接传递数据在程序开发中较为常用，它可以代替按钮来实现某种功能，本节主要讲解通过超链接传递数据、接收超链接传递的数据，以及对超链接传递的数据进行编码和解码。

6.3.1　通过超链接传递数据

定义超链接使用<a>标记，其语法格式如下：

```
<a href=URL name=name target=target>链接文字</a>
```

<a>标记的属性介绍如下。

- ☑ href 属性：指定所链接文件的 URL 路径。这个路径可以是相对路径，也可以是绝对路径。
- ☑ name 属性：指定页面的锚点名称，如果需要链接到对应的锚点位置，需要在锚点名称前添加一个“#”字符。
- ☑ target 属性：指定要打开的链接所使用的浏览器窗口名称，可以使用自定义的窗口名称，也可以使用下面 4 个内置的窗口名称，其中前两个比较常用。
  - ➢ _self：在当前窗口中打开链接文件，是默认值。
  - ➢ _blank：开启一个新的窗口打开链接文件。
  - ➢ _parent：在父级窗口中打开文件，常用于框架页面。
  - ➢ _top：在顶层窗口中打开文件，常用于框架页面。

下面是一段定义锚点和超链接的示例代码，使用文字链接和图片链接，以及通过相对 URL 和绝对 URL 分别链接本地文件和远程文件。

例 6.05　下面分别创建文字和图片超链接，代码如下。（**实例位置：光盘\mr\06\sl\6.05**）

```
<body>
<div align="center"><a target="_blank" href="index_ok.php?name=<?php echo "你好啊！";?>">点我呀！
</a>   //创建文字链接
  <label><a   target="_blank"  href="index_ok.php?photo=<?php  echo  "明日科技全体员工向您拜年!";?>"><img src="images/2010.jpg" width="687" height="177" border="0" /></a></label>
              //创建图片链接
</div>
</body>
```

上面代码中定义了“点我呀！”超链接，并设置链接跳转页为 index_ok.php 页面，传值为输出“你好啊！”字符串。运行结果如图 6.6 所示。

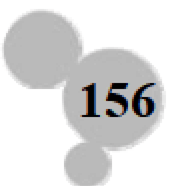


图 6.6　通过超链接传递数据

6.3.2　通过$_GET[]获取超链接传递的数据

获取超链接传递的数据，需要使用$_GET[]方法获取。

例 6.06　本例通过$_GET[]接收例 6.05 中超链接传递的数据，代码如下。（**实例位置：光盘\mr\06\sl\6.06**）



```
<?php
if(isset($_GET['name'])){                          //判断文字链接是否存在
    echo $_GET['name'];                            //输出文字链接内容
}
if(isset($_GET['photo'])){                         //判断图片链接是否存在
    echo $_GET['photo'];                           //输出图片链接内容
}
?>
```

执行程序，运行结果如图 6.7 和图 6.8 所示。



图 6.7　获取文字链接传递的数据

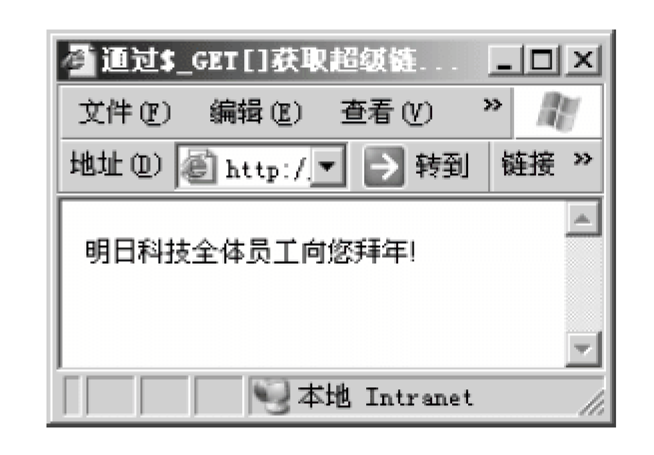


图 6.8　获取图片链接传递的数据

6.3.3　对超链接传递的数据进行编码、解码

1．对超链接传递的数据进行编码

使用 URL 参数传递数据，就是在 URL 地址后面加上适当的参数。URL 实体对这些参数进行处理。使用方法如下：

```
http://url?name1=value1&name2=value2……
```

URL 传递的参数，也称为查询字符串

显而易见，这种方法会将参数暴露无疑，因此，本节针对该问题讲述一种 URL 编码方式，对 URL 传递的参数进行编码。

URL 编码是一种浏览器用来打包表单输入数据的格式，是对用地址栏传递参数进行的一种编码规则。例如，在参数中带有空格，则传递参数时就会发生错误，而用 URL 编码过以后，空格转换成了“%20”，这样错误就不会发生；对中文进行编码也是同样的情况，最主要的一点就是它可以对 URL 传递的参数进行编码。

PHP 中对字符串进行 URL 编码使用的是 urlencode()函数，该函数的语法如下：

```
string urlencode( string str)
```

该函数可以实现将字符串 str 进行 URL 编码。

说明：对于服务器而言，编码前后的字符串并没有什么区别，服务器能够自动识别。这里是为了讲解 URL 编码的使用方法，而在实际应用中，对一些非保密性的参数不需要进行编码，读者可根据实际情况有选择地使用。

2．对超链接传递的数据进行解码

对于 URL 传递的参数直接应用$_GET[ ]方法获取即可。而对于进行 URL 加密的查询字符串，需要通过 urldecode()函数对获取后的字符串进行解码。该函数的语法如下：

```
string urldecode( string str)
```

该函数可以实现将 URL 编码 str 查询字符串进行解码。

6.4　了解 JavaScript 脚本

JavaScript 脚本语言与其他语言一样，有其自身的基本数据类型、表达式和运算符以及程序的基本框架结构。通过本节的学习，读者可以掌握更多 JavaScript 脚本语言的基础知识。

6.4.1　JavaScript 语言基础

1．JavaScript 能做什么

JavaScript 是比较流行的一种制作网页特效的脚本语言，它由客户端浏览器解释执行，可以应用在 PHP、ASP、JSP 和 ASP.NET 网站中，同时目前比较热门的 Ajax 就是以 JavaScript 为基础。由此可见，熟练掌握并应用 JavaScript 对于网站开发人员非常重要。

JavaScript 主要应用于以下几个方面：

（1）在网页中加入 JavaScript 脚本代码，可以使网页具有动态交互的功能，便于网站与用户间的沟通，及时响应用户的操作，对提交的表单做即时的检查，如验证表单元素是否为空、验证表单元素是否是数值型、检测表单元素是否输入错误等。

（2）应用 JavaScript 脚本制作网页特效，如动态的菜单、浮动的广告等，为页面增添绚丽的动态效果，使网页内容更丰富、更活泼。

（3）应用 JavaScript 脚本建立复杂的网页内容，如打开新窗口载入网页。

（4）应用 JavaScript 脚本可以对用户的不同事件产生不同的响应。

（5）应用 JavaScript 制作各种各样的图片、文字、鼠标、动画和页面的效果。

（6）应用 JavaScript 制作一些小游戏。

2．JavaScript 数据类型

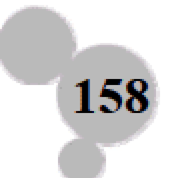
JavaScript 主要有 6 种数据类型，如表 6.5 所示。

表 6.5　JavaScript 数据类型

数 据 类 型	说　　明	举　　例
字符串型	使用单引号或双引号括起来的一个或多个字符	如"PHP"、"I like study PHP"等
数值型	包括整数或浮点数（包含小数点的数或科学计数法的数）	如-128、12.9、6.98e6 等
布尔型	布尔型常量只有两种状态，即 TRUE 或 FALSE	如 event.returnValue=FALSE
对象型	用于指定 JavaScript 程序中用到的对象	如网页表单元素
null 值	可以通过给一个变量赋 null 值来清除变量的内容	如 a=null
undefined	表示该变量尚未被赋值	如 var a

3．JavaScript 变量

变量是指程序中一个已经命名的存储单元，它的主要作用就是为数据操作提供存放信息的容器。对于变量的使用首先必须明确变量的命名规则、变量的声明与赋值。



（1）变量的命名规则

JavaScript 变量的命名规则如下：

☑ 必须以字母或下划线开头，中间可以是数字、字母或下划线。

☑ 变量名不能包含空格或加号、减号等符号。

☑ JavaScript 的变量名是严格区分大小写的。例如，User 与 user 代表两个不同的变量。

☑ 不能使用 JavaScript 中的关键字。JavaScript 的关键字如表 6.6 所示。

表 6.6　JavaScript 的关键字

关　键　字					
abstract	continue	finally	instanceof	private	this
boolean	default	float	int	public	throw
break	do	for	interface	return	typeof
byte	double	function	long	short	TRUE
case	else	goto	native	static	var
catch	extends	implements	new	super	void
char	FALSE	import	null	switch	while
class	final	in	package	synchronized	with

说明： 虽然 JavaScript 的变量可以任意命名，但为了在编程时使代码更加规范，最好使用便于记忆且有意义的变量名称，以增加程序的可读性。

（2）变量的声明与赋值

在 JavaScript 中，一般在使用变量前需要先声明变量，不过有时变量可以不先声明，使用时再根据变量的实际作用来确定其所属的数据类型。所有的 JavaScript 变量都由关键字 var 声明。其语法如下：

```
var variable;
```

在声明变量的同时也可以对变量进行赋值：

```
var variable=08;
```

技巧： 建议在使用变量前就对其声明，因为声明变量的最大好处就是能及时发现代码中的错误。由于 JavaScript 是采用动态编译的，而动态编译不易于发现代码中的错误，特别是变量命名方面的错误。

声明变量时所遵循的规则如下。

☑ 可以使用一个关键字 var 同时声明多个变量，例如：

```
var m,n;
```

☑ 可以在声明变量的同时对其赋值，即为初始化，例如：

```
var m=1;n=100;
```

☑ 如果只是声明了变量，并未对其赋值，则默认值为 undefined。

注意： 在 JavaScript 中，可以使用分号代表一个语句的结束，如果每个语句都在不同的行中，那么分号可以省略；如果多个语句在同一行中，分号就不能省略。建议读者不要省略分号，以养成良好的编程习惯。

例如，声明 3 个不同数据类型的变量，代码如下：

```
var i=100;                              //定义变量 i 为数值类型
var str="明日科技，明天会更好！";          //定义变量 str 为字符串类型
var content=TRUE;                       //定义变量 content 为布尔类型
```

注意：在程序开发过程中，可以使用 var 语句多次声明同一个变量，如果重复声明的变量已经有一个初始值，那么此时的声明变量就相当于对变量重新赋值。

4. JavaScript 注释

在 JavaScript 中，采用的注释方法有以下两种。

☑ 单行注释

单行注释使用“//”进行标识。“//”符号后面的文字都不被程序解释执行。例如：

```
//这里是程序代码的注释
```

☑ 多行注释

多行注释使用“/*…*/”进行标识。“/*…*/”符号后面的文字不被程序解释执行。例如：

```
/*
这里是多行程序注释
*/
```

注意：多行注释“/*…*/”中可以嵌套单行注释“//”，但是不可以嵌套多行注释“/*…*/”。因为第一个“/*”会与其后面第一个“*/”相匹配，从而使后面的注释不起作用，甚至导致程序出错。

另外，JavaScript 还能识别 HTML 注释的开始部分“<!--”，JavaScript 会将其看作单行注释结束，如使用“//”一样。但是 JavaScript 不能识别 HTML 注释的结果部分“-->”。

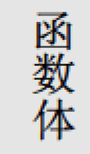
这种现象存在的主要原因是：在 JavaScript 中，如果第一行以“<!--”开始，最后一行以“-->”结束，那么其间的程序就包含在一个完整的 HTML 注释中，会被不支持 JavaScript 的浏览器忽略掉，不能被显示。如果第一行以“<!--”开始，最后一行以“//-->”结束，JavaScript 会将两行都忽略掉，而不会忽略这两行之间的部分。用这种方式可以针对那些无法理解 JavaScript 的浏览器而隐藏代码，而对那些可以理解 JavaScript 的浏览器则不必隐藏。

6.4.2 通过 JavaScript 脚本自定义函数

自定义函数就是由用户自己命名并且书写的能实现特定功能的程序单元。用户使用自定义函数必须事先声明，不能直接使用未声明的自定义函数。

JavaScript 用 function 来定义函数，语法格式如下：

```
function 函数名([参数]){
    return var;
}
```



函数体

自定义函数的调用方法是：

```
函数名();
```

其中的括号绝对不能省略。

例 6.07 下面创建 JavaScript 脚本自定义 check()函数，具体代码如下。**（实例位置：光盘\mr\06\sl\6.07）**

```
<script language="javascript">
function check(){                  //自定义函数
    var a=10;                      //定义变量
    if(a>0){                       //判断变量是否大于 0
```

```
            return TRUE;                //返回 TRUE
        }
            return FALSE;               //返回 FALSE
}
</script>
```

这段代码中的 return 代表函数的返回值，当 a>0 时返回 TRUE，否则返回 FALSE。

注意：在同一个页面中不能定义名称相同的函数。另外，当用户自定义函数后，需要对该函数进行引用，否则自定义的函数将失去意义。

6.4.3　JavaScript 脚本嵌入方式

1. 在 HTML 中嵌入 JavaScript 脚本

JavaScript 作为一种脚本语言，可以嵌入到 HTML 文件中。在 HTML 中嵌入 JavaScript 脚本的方法是使用<script>标记。其语法如下：

```
<script   language="javascript">
…
</script>
```

应用<script>标记是直接执行 JavaScript 脚本最常用的方法，大部分含有 JavaScript 的网页都采用这种方法，其中，通过 language 属性可以设置脚本语言名称和版本。

注意：如果在<script>标记中未设置 language 属性，Internet Explorer 浏览器和 Netscape 浏览器将默认使用 JavaScript 脚本语言。

例 6.08　实现在 HTML 中嵌入 JavaScript 脚本，直接在<script>和</script>标识符中间写入 JavaScript 代码，用于弹出一个提示对话框，代码如下。（**实例位置：光盘\mr\06\sl\6.08**）

```
<html>
<head>
<title>在 HTML 中嵌入 JavaScript 脚本</title>
</head>
<body>
<script language="javascript">
    alert("PHP 编程词典!");
</script>
</body>
</html>
```

在上面的代码中，<script>与</script>标记之间调用 JavaScript 脚本语言 window 对象的 alert 方法向客户端浏览器弹出一个提示对话框。在 IE 浏览器中打开 HTML 文件，运行结果如图 6.9 所示。



图 6.9　在 HMTL 中嵌入 JavaScript 脚本

注意：JavaScript 脚本通常写在<head>…</head>标记和<body>…</body>标记之间。写在<head>标记中间的一般是函数和事件处理函数，写在<body>标记中间的是网页内容或调用函数的程序块。

在 HTML 中通过“javascript:”可以调用 JavaScript 的方法。例如，在页面中插入一个按钮，在该按钮的 onClick 事件中应用“javascript:”调用 window 对象的 alert 方法弹出一个警告提示框，代码如下：

```
<input type="submit" name="Submit" value="单击这里" onClick="javascript:alert('您单击了这个按钮！')">
```

2. 应用 JavaScript 事件调用自定义函数

在 Web 程序开发过程中，经常需要在表单元素的相应事件下调用自定义函数。例如，在按钮的单击事件下调用自定义函数 check()，验证表单元素是否为空，代码如下：

```
<input type="submit" name="Submit" value="检测" onClick="check();">
```

然后在该 Web 表单的当前页写一个 check()自定义函数即可。

例 6.09 创建用户登录页面，在用户登录按钮中通过 onClick 事件调用自定义函数，验证表单中提交的元素是否为空。自定义函数的代码如下。**（实例位置：光盘\mr\06\sl\6.09）**

```
<script>
function check_form(){
	if(form1.username.value==""){
		alert("请输入用户名！");
			form1.username.select();
			return(FALSE);
		}
	if(form1.password.value==""){
		alert("请输入密码！");
		form1.password.select();
		return(FALSE);
		}
}
</script>
```

创建 form 表单元素的代码如下：

```
<form id="form1" name="form1" method="post" action="index.php">
<input name="username" type="text" id="username" size="18" />
<input name="password" type="password" id="password" size="18" />
<input type="submit" name="Submit" value="登录" onclick="return check_form();" />
<input type="reset" name="Submit2" value="重置" />
</form>
```

如果在登录页面中直接单击“登录”按钮，将弹出如图 6.10 所示的页面。



图 6.10 应用 JavaScript 事件调用自定义函数

3. 在 PHP 动态网页中引用 JS 文件

在网页中，除了在<script>与</script>标记之间编写 JavaScript 脚本代码以外，还可以通过<script>标记中的 src 属性指定外部的 JavaScript 文件（即 JS 文件，以.js 为扩展名）的路径，从而引用对应的 JS 文件。



其语法如下：

```
<script src=url language="Javascript"></script>
```

其中，url 是 JS 文件的路径，其中“language="Javascript"”可以省略，因为<script>标记默认使用的就是 JavaScript 脚本语言。

JavaScript 脚本不仅可以与 HTML 结合在一起使用，同时也可以与 PHP 动态网页结合使用，其引用的方法是相同的。使用外部 JS 文件的优点如下：

☑ 使用 JS 文件可以将 JavaScript 脚本代码从网页中独立出来，便于代码的阅读。

☑ 一个外部 JS 文件，可以同时被多个页面调用。当公共的 JavaScript 脚本代码需要修改时，只需要修改一个 JS 文件中的代码即可，便于代码的维护。

☑ 通过<script>标记中的 src 属性不但可以调用同一个服务器上的 JS 文件，还可以通过指定路径来调用其他服务器上的 JS 文件。

例 6.10　在网页中，通过<script>标记的 src 属性引用外部的 JS 文件，用于弹出一个提示对话框。index.php 文件中的代码如下。**（实例位置：光盘\mr\06\sl\6.10）**

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>通过 script 标记的 src 属性引用外部的 JS 文件</title>
</head>
<script language="javascript" src="js/script.js"></script>
<body>
<table width="1149" height="342" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td width="1149"><img src="images/bg.JPG" width="1147" height="813"></td>
  </tr>
</table>
</body>
</html>
```

将 script.js 文件存储在根目录下的 js 文件夹中，代码如下：

```
var newformW=300;
var newformH=180;
function pp(){
	var T=window.screen.width-newformW;
	var L=window.screen.height+newformH;
	var
name=window.open("advertise.htm","","width="+newformW+",height="+newformH+",top="+T+",left="+L);
}
pp();
```

从上面代码中可以看出，在 index.php 文件中通过设定<script>标记中的 src 属性，引用 js 文件夹下的 script.js 文件。在 script.js 文件中调用 JavaScript 脚本 window 对象的 open 方法打开一个指定的 html 文件，并设置该页面的大小，以及在主页面中的位置。运行结果如图 6.11 所示。

在网页中引用 JS 文件的注意事项如下：

☑ 在 JS 文件中，只能包含 JavaScript 脚本代码，不能包含<script>标记和 HTML 代码。

☑ 在引用 JS 文件的<script>与</script>标记之间不应存在其他的 JavaScript 代码，即使存在 JavaScript 代码，浏览器也会忽略此脚本代码，而只执行 JS 文件中的 JavaScript 脚本代码。



图 6.11　在 PHP 动态网页中引用 JS 文件

4．解决浏览器不支持 JavaScript 问题

虽然大多数浏览器都支持 JavaScript 脚本，但是由于浏览器设置的不同，导致很多浏览器不支持 JavaScript 脚本。如果遇到不支持 JavaScript 脚本的浏览器，网页会达不到预期效果或出现错误。解决这个问题可以使用以下 4 种方法。

（1）开启 IE 浏览器对 JavaScript 的支持。

目前有些支持 JavaScript 的浏览器为了安全关闭了对 JavaScript 的支持。这时，浏览者可以启用对 JavaScript 脚本的支持来解决这一问题。具体操作步骤如下：

① 选择 IE 浏览器菜单中的“工具”/“Internet 选项”命令，打开“Internet 选项”对话框，选择“安全”选项卡，选择 Internet 安全设置项，单击“自定义级别”按钮，打开如图 6.12 所示的对话框。

② 将“Java 小程序脚本”和“活动脚本”两个选项设置为启用状态。单击“确定”按钮，即可开启 IE 浏览器支持 JavaScript 脚本的功能。

（2）开启 IE 浏览器对本地 JavaScript 的支持。

IE 浏览器将网页分为 Internet、本地 Intranet、受信任的站点和受限制的站点 4 个区域，但不包括本地网页。通常在 Windows XP 操作系统中，打开 IE 浏览器中包含 JavaScript 的网页时，弹出如图 6.13 所示的安全提示对话框。

如果用户要继续执行网页中包含的 JavaScript 脚本，可以用鼠标右键单击安全提示区域，在弹出的快捷菜单中选择“允许阻止的内容”命令，如图 6.13 所示，即可成功运行本网站。但此选项仅针对当前网页。若要永久消除 IE 浏览器的这种安全提示，需要对 IE 浏览器进行如下设置。

选择 IE 浏览器菜单中的“工具”/“Internet 选项”命令，打开“Internet 选项”对话框，选择“高级”选项卡，在安全设置区选中“允许活动内容在我的计算机上的文件中运行”和“允许来自 CD 的活动内容在我的计算机上运行”复选框（此选项仅适用于 Windows XP 操作系统），单击“确定”按钮，即可成功解决上述问题。

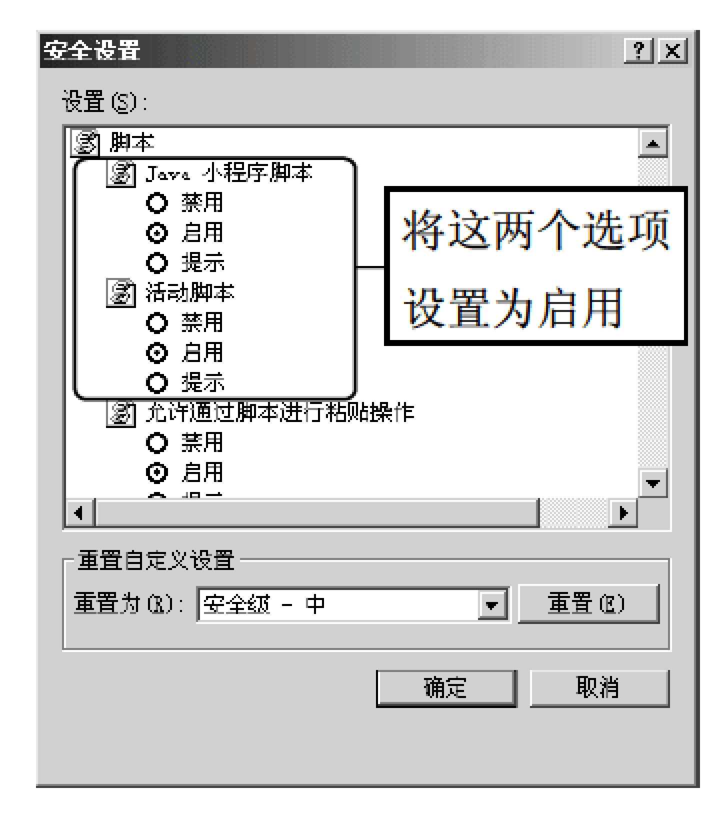


图 6.12　启用 JavaScript 脚本功能

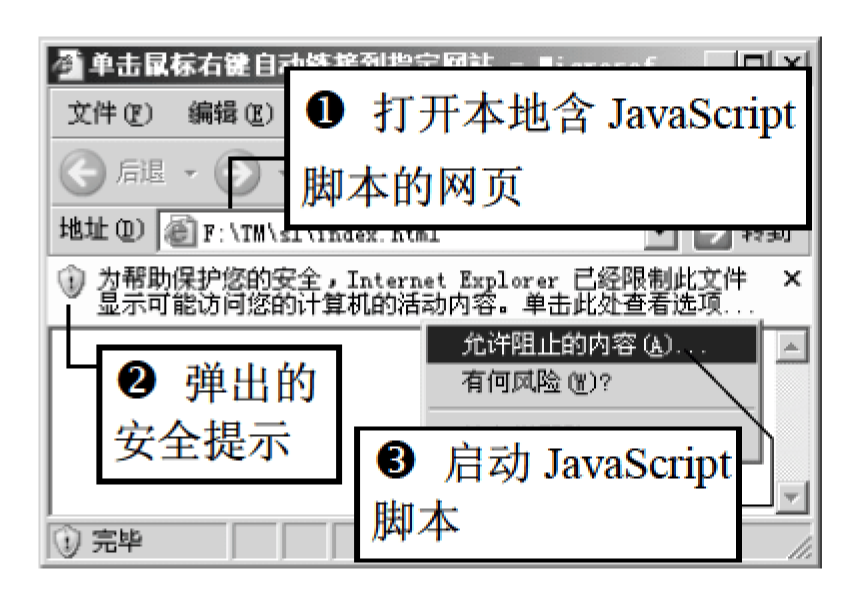


图 6.13　安全提示对话框

（3）应用注释符号验证浏览器是否支持 JavaScript 脚本功能。

如果用户不能确定浏览器是否支持 JavaScript 脚本，那么可以应用 HTML 提供的注释符号进行验证。HTML 注释符号是以“<!--”开始以“-->”结束的。如果在此注释符号内编写 JavaScript 脚本，对于不支持 JavaScript 的浏览器，将会把编写的 JavaScript 脚本作为注释处理。

例 6.11　使用 JavaScript 脚本在页面中输出一个字符串，将 JavaScript 脚本编写在 HTML 注释中，如果浏览器支持 JavaScript 将输出此字符串，如果不支持将不输出此字符串，代码如下。（**实例位置：光盘\mr\06\sl\6.11**）

```
<html>
<head>
<title>应用注释符号验证浏览器是否支持 JavaScript 脚本功能</title>
</head>
<body>
<script type="text/javascript" >
<!--
    document.write("您的浏览器支持 JavaScript 脚本！");
-->
</script>
</body>
</html>
```

在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 6.14 所示。



图 6.14　应用注释符号验证浏览器是否支持 JavaScript 脚本功能

（4）应用<noscript>标记验证浏览器是否支持 JavaScript 脚本。

除了可以通过上述的应用注释符号验证浏览器是否支持 JavaScript 脚本外，还可以应用<noscript>标记进行验证。

如果当前浏览器支持 JavaScript 脚本，那么该浏览器将会忽略<noscript>…</noscript>标记之间的任何内

容。如果浏览器不支持 JavaScript 脚本，那么浏览器将会把<noscript>…</noscript>标记之间的内容显示出来。此标记可以提醒浏览者当前使用的浏览器是否支持 JavaScript 脚本。

例 6.12 使用 JavaScript 脚本在页面中输出一个字符串，并使用<noscript>标记提醒浏览者当前浏览器是否支持 JavaScript 脚本，代码如下。**（实例位置：光盘\mr\06\sl\6.12）**

```
<html>
<head>
<title>应用&lt;noscript&gt;标记验证浏览器是否支持 JavaScript 脚本</title>
</head>
<body>
<script language="javascript">
    document.write("您的浏览器支持 JavaScript 脚本");
</script>
<noscript>
    您的浏览器不支持 JavaScript 脚本
</noscript>
</body>
</html>
```

在 IE 浏览器中输入地址，按 Enter 键，运行结果如图 6.15 所示。



图 6.15 应用<noscript>标记验证浏览器是否支持 JavaScript 脚本

技巧： 当解释程序遇到</script>标记时会终止当前脚本。要显示“</script>”本身，将“<”改写为“<”，将“>”改写为“>”即可。若要使用 document.write 输出<script></script>标签时，需要将闭合标签通过反斜杠进行转义，如<script>…<\/script>。

6.4.4 JavaScript 常用事件

为了便于读者查找 JavaScript 中的常用事件，下面以表格的形式对各事件进行说明，如表 6.7 所示。

表 6.7 JavaScript 中的常用事件

状　态	事　件	说　明
鼠标键盘事件	onClick	鼠标单击时触发此事件
	onDblClick	鼠标双击时触发此事件
	onMouseDown	按下鼠标时触发此事件
	onMouseUp	鼠标按下后松开鼠标时触发此事件
	onMouseOver	当鼠标移动到某对象范围的上方时触发此事件
	onMouseMove	鼠标移动时触发此事件
	onMouseOut	当鼠标离开某对象范围时触发此事件
	onKeyPress	当键盘上的某个按键被按下并且释放时触发此事件
	onKeyDown	当键盘上的某个按键被按下时触发此事件
	onKeyUp	当键盘上的某个按键被按下后松开时触发此事件

续表

状　　态	事　　件	说　　明
页面相关事件	onAbort	图片在下载时被用户中断时触发此事件
	onLoad	页面内容完成时触发此事件（也就是页面加载事件）
	onResize	当浏览器的窗口大小被改变时触发此事件
	onUnload	当前页面将被改变时触发此事件
表单相关事件	onBlur	当前元素失去焦点时触发此事件
	onChange	当前元素失去焦点并且元素的内容发生改变时触发此事件
	onFocus	当某个元素获得焦点时触发此事件
	onReset	当表单中 reset 的属性被激活时触发此事件
	onSubmit	一个表单被递交时触发此事件
滚动字幕事件	onBounce	在 Marquee 内的内容移动至 Marquee 显示范围之外时触发此事件
	onFinish	当 Marquee 元素完成需要显示的内容后触发此事件
	onStart	当 Marquee 元素开始显示内容时触发此事件

6.5　照猫画虎——基本功训练

6.5.1　基本功训练 1——设计一个用户登录表单

视频讲解：光盘\mr\06\lx\设计一个用户登录表单.exe

实例位置：光盘\mr\06\zmhh\01

本例使用 Dreamweaver 开发工具设计一个用户登录界面，运行结果如图 6.16 所示。

具体步骤如下：

（1）设计用户登录页面，效果如图 6.17 所示。



图 6.16　用户登录界面



图 6.17　设计用户登录界面

（2）创建一个表单，在表单中添加两个元素，即用户名（name）和密码（pwd），以及两个图像域，即“登录”和“取消”，代码如下：

```
<form id="form1" name="form1" method="post" action="">
  <table id="__01" width="452" height="255" border="0" cellpadding="0" cellspacing="0">
    <tr>
      <td colspan="7"><img src="images/dl_01.gif" width="452" height="81" alt="" /></td>
    </tr>
    <tr>
```

```
        <td rowspan="3"><img src="images/dl_02.gif" width="59" height="173" alt="" /></td>
        <td height="107" colspan="4" align="center" bgcolor="#FEF5CC"><table width="285" height="43">
          <tr>
            <td width="96">用户名：</td>
            <td width="177"><label>
            <input type="text" name="name" />
            </label></td>
          </tr>
          <tr>
            <td>密码：</td>
            <td><label>
              <input type="password" name="pwd" />
            </label></td>
          </tr>
        </table></td>
        <td colspan="2" rowspan="2"><img src="images/dl_04.gif" width="58" height="141" alt="" /></td>
      </tr>
      <tr>
        <td><img src="images/dl_05.gif" width="92" height="34" alt="" /></td>
        <td width="78" height="34"><input type="image" name="imageField" src="images/dl_06.gif" /></td>
        <td width="83" height="34"><input type="image" name="imageField2" src="images/dl_07.gif" /></td>
        <td><img src="images/dl_08.gif" width="82" height="34" alt="" /></td>
      </tr>
      <tr>
        <td colspan="5"><img src="images/dl_09.gif" width="358" height="32" alt="" /></td>
        <td><img src="images/dl_10.gif" width="35" height="32" alt="" /></td>
      </tr>
      <tr>
        <td><img src="images/分隔符.gif" width="59" height="1" alt="" /></td>
        <td><img src="images/分隔符.gif" width="92" height="1" alt="" /></td>
        <td><img src="images/分隔符.gif" width="78" height="1" alt="" /></td>
        <td><img src="images/分隔符.gif" width="83" height="1" alt="" /></td>
        <td><img src="images/分隔符.gif" width="82" height="1" alt="" /></td>
        <td><img src="images/分隔符.gif" width="23" height="1" alt="" /></td>
        <td><img src="images/分隔符.gif" width="35" height="1" alt="" /></td>
      </tr>
    </table>
</form>
```

照猫画虎：根据上面的实例，设计一个用户注册表单页面。（20 分）（实例位置：光盘\mr\06\zmhh\01_zmhh）

6.5.2 基本功训练 2——验证用户名和密码不能为空

视频讲解：光盘\mr\06\lx\验证用户名和密码不能为空.exe

实例位置：光盘\mr\06\zmhh\02

下面制作一个简单的用户登录界面，并且验证用户名和密码不能为空，如果为空则给出提示。运行结果如图 6.18 所示。

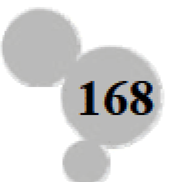




图 6.18　提示请输入用户名

具体步骤如下：

（1）设计登录页面，效果如图 6.18 所示。

（2）通过 JavaScript 脚本判断用户和密码是否为空，具体代码如下：

```
<script language="javascript">
function checkit(){                          //自定义函数
        if(form1.name.value==""){            //判断用户名是否为空
                alert("请输入用户名!");
                 form1.name.select();
                  return FALSE;
            }
            if(form1.pwd.value==""){         //判断密码是否为空
                  alert("请输入密码!");
                  form1.pwd.select();
                  return FALSE ;
            }
                  return TRUE;

}
</script>
```

（3）通过“登录”按钮的 onclick 事件调用自定义函数 checkit()，代码如下：

```
<input type="image" name="imageField" onclick="return checkit();" src="images/dl_06.gif" />
      //登录按钮事件
<input type="image" name="imageField2" onclick="form.reset(); return FALSE;" src="images/dl_07.gif" />
      //取消按钮事件
```

照猫画虎：验证用户名是否以指定字符串开头。提示：应用 substr()函数获取文本框值中的第一个字符，然后通过自定义函数 check()判断获取的第一个字符是否等于“M”。（20 分）（实例位置：光盘\mr\06\zmhh\02_zmhh）

6.5.3　基本功训练 3——输出用户登录信息

视频讲解：光盘\mr\06\lx\输出用户登录信息.exe

实例位置：光盘\mr\06\zmhh\03

下面制作一个简单的用户登录界面，当用户单击“登录”按钮后，在页面中显示用户输入的用户名和密码。运行结果如图 6.19 所示。

图 6.19　输出用户登录信息

具体步骤如下：

（1）设计登录页面，效果如图 6.19 所示。

（2）通过 JavaScript 脚本判断用户和密码是否为空，具体代码如下：

```
<script language="javascript">
function checkit(){                         //自定义函数
      if(form1.name.value==""){             //判断用户名是否为空
              alert("请输入用户名!");
               form1.name.select();
                return FALSE;
          }
        if(form1.pwd.value==""){            //判断密码是否为空
                alert("请输入密码!");
                form1.pwd.select();
                return FALSE ;
            }
                return TRUE;

}
</script>
```

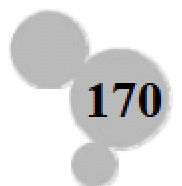
（3）通过“登录”按钮的 onclick 事件调用自定义函数 checkit()，代码如下：

```
<input type="image" name="imageField" onclick="return checkit();" src="images/dl_06.gif" />
      //登录按钮事件
<input type="image" name="imageField2" onclick="form.reset(); return FALSE;" src="images/dl_07.gif" />
      //取消按钮事件
```

（4）输入用户名和密码，单击“登录”按钮，在页面中即可打印出输入的用户名和密码，具体代码如下：

```
<?php
if(isset($_POST['name']) && $_POST['pwd']!=""){          //判断用户名是否存在
      echo "用户名为：".$_POST['name']."<br>";            //输出用户名
      echo "密码为：".$_POST['pwd'];                      //输出密码
}
?>
```

照猫画虎：根据上面的实例，判断用户输入的用户名和密码是否正确，如果正确则提示登录成功，否则提示登录失败。提示：将用户名和密码定义为变量。（20 分）（实例位置：光盘\mr\06\zmhh\03_zmhh）



6.5.4　基本功训练 4——验证 email 地址格式是否正确

视频讲解：光盘\mr\06\lx\验证 email 地址格式是否正确.exe

实例位置：光盘\mr\06\zmhh\04

本例通过 JavaScript 脚本和正则表达式来验证 email 地址格式，运行结果如图 6.20 所示。



图 6.20　验证 email 地址格式

验证 email 地址的正则表达式如下：

```
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/
```

具体步骤如下：

（1）使用 JavaScript 编写一个用于检测 email 地址是否正确的函数 checkemail()，该函数只有一个参数 email，用于获取输入的 email 地址，返回值为 TRUE 或 FALSE。代码如下：

```
<script language="javascript">
function checkemail(email){
	var str=email;
	 //在 JavaScript 中，正则表达式只能使用"/"开头和结束，不能使用双引号
	var Expression=/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;
	var objExp=new RegExp(Expression);
	if(objExp.test(str)==TRUE){
		return TRUE;
	}else{
		return FALSE;
	}
}
</script>
```

（2）调用 checkemail()函数判断 email 地址是否正确，并显示相应的提示信息。关键代码如下：

```
<script language="javascript">
function check(form1){
	if(form1.email.value==""){
		alert("请输入 email 地址!");form1.email.focus();return;
	}
```

```
        if(!checkemail(form1.email.value)){
            alert("您输入的 email 地址不正确!");form1.email.focus();return;
        }
        form1.submit();
}
</script>
```

照猫画虎：根据上面的例子，验证电话号码是否正确。提示：验证电话号码的正则表达式是“/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/;”。（20 分）（实例位置：光盘\mr\06\zmhh\04_zmhh）

6.5.5　基本功训练 5——验证用户两次输入的密码是否一致

视频讲解：光盘\mr\06\lx\验证用户两次输入的密码是否一致.exe

实例位置：光盘\mr\06\zmhh\05

本例使用 JavaScript 脚本实现判断用户两次输出的密码是否一致，不一致时弹出提示，运行结果如图 6.21 所示。



图 6.21　判断用户两次输入的密码是否一致

具体步骤如下：

（1）编写 JavaScript 脚本，判断两次输入的密码是否一致，代码如下：

```
<script language="javascript">
function checkit(){                                        //自定义函数
    if(form1.pwd.value!=form1.qrpwd.value){                //判断密码是否一致
            alert("两次输入的密码不一致！请重新输入！");
            form1.name.select();
            return FALSE;
        }
            return TRUE;

}
</script>
```

（2）通过“注册”按钮的 onclick 事件调用自定义函数 checkit()，代码如下：

```
<input type="image" name="imageField" onclick="return checkit();"   src="images/zc_06.gif" />
```

照猫画虎：验证身份证号码是否正确。提示：正则表达式格式为“/^\d{17}[\d|X]|\d{15}/”。（20 分）（实例位置：光盘\mr\06\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

6.6　情景应用——拓展与实践

6.6.1　情景应用 1——开发一个用户注册表单

视频讲解：光盘\mr\06\lx\开发一个用户注册表单.exe

实例位置：光盘\mr\06\qjyy\01

本例通过表单元素创建一个用户注册页面，运行结果如图 6.22 所示。

图 6.22　用户注册页面

具体步骤如下：

（1）设计用户注册页面，效果如图 6.22 所示。

（2）实现用户注册操作，当用户单击“注册”按钮时，判断信息是否符合标准，如果符合则可成功注册，并将信息保存到数据库中。代码如下：

```
<?php
//连接数据库
```

```
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database06",$conn);
mysql_query("set names utf8");
$name=$_POST['name'];
$pwd=$_POST['pwd'];
$qpwd=$_POST['qpwd'];
$sex=$_POST['sex'];
$mail=$_POST['email'];
$tel=$_POST['tel'];
if(isset($_POST['Submit']) and $_POST['Submit']=="注册"){                    //判断用户是否单击“注册”按钮
	if($name!=null){                                                          //判断用户名是否为空
		if($pwd==$qpwd){                                                      //判断两次密码是否一致
			if(preg_match("/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/",$email,$counts)){  //判断 email 地址
				$insert=mysql_query("insert        into        tb_login(name,pwd,qpwd,sex,email,tel)
values('$name','$pwd','$qpwd','$sex','$mail','$tel')",$conn);
					if($insert){
						echo "<script> alert('注册成功!'); window.location.href='index.php'</script>";
					}else{
						echo "<script> alert('注册失败!'); window.location.href='index.php'</script>";
					}
				}else{
				echo "<script> alert('email 地址不正确!'); window.location.href='index.php'</script>";
			}
		}else{
			echo "<script> alert('密码不一致!'); window.location.href='index.php'</script>";
		}
	}else{
		echo "<script> alert('请输入用户名!'); window.location.href='index.php'</script>";
	}
}
?>
```

DIY：实现同一个页中的多表单提交，在一个页面中添加多个表单，当用户提交表单时判断用户提交的是哪个表单。（20 分）（实例位置：光盘\mr\06\qjyy\01_diy）

6.6.2 情景应用 2——验证用户注册信息是否合理

视频讲解：光盘\mr\06\lx\验证用户注册信息是否合理.exe

实例位置：光盘\mr\06\qjyy\02

本例使用 JavaScript 脚本实现验证用户注册信息是否合理，当信息不合理时给出相应的提示，运行结果如图 6.23 所示。

具体步骤如下：

（1）创建 index.php 文件，设计用户注册信息页面，效果如图 6.23 所示。

（2）编写 JavaScript 脚本文件，完成验证用户注册信息是否合理，代码如下：

```
<script language="javascript">
function checkemail(email){
	var str=email;
	 //在 JavaScript 中，正则表达式只能使用"/"开头和结束，不能使用双引号
```

```
        var Expression=/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;     //定义 email 地址格式
        var objExp=new RegExp(Expression);
        if(objExp.test(str)==TRUE){
                return TRUE;
        }else{
                return FALSE;
        }
}
function checktel(tel){
        var str=tel;
         //在 JavaScript 中，正则表达式只能使用"/"开头和结束，不能使用双引号
        var Expression=/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/;                  //定义电话格式
        var objExp=new RegExp(Expression);
        if(objExp.test(str)==TRUE){
                return TRUE;
        }else{
                return FALSE;
        }
}
function checkit(){                                                          //自定义函数
        if(form1.name.value==""){                                            //判断用户名是否为空
                    alert("请输入用户名!");
                     form1.name.select();
                      return FALSE;
                }
        if(form1.pwd.value==""){                                             //判断密码是否为空
                    alert("请输入密码!");
                     form1.pwd.select();
                      return FALSE;
                }
        if(form1.qpwd.value==""){                                            //判断确认密码是否为空
                    alert("请输入确认密码!");
                     form1.qpwd.select();
                      return FALSE;
                }
        if(form1.pwd.value!=form1.qpwd.value){                               //判断两次输入的密码是否一致
                      alert("两次输入的密码不一致!");
                      form1.pwd.select();
                      return FALSE ;
                  }
        if(form1.email.value==""){                                           //判断 email 是否为空
                    alert("请输入 email!");
                     form1.email.select();
                      return FALSE;
                }
        if(!checkemail(form1.email.value)){                                  //判断 email 地址格式是否正确
                      alert("您输入 email 地址不正确!");
                      form1.email.select();
                      return FALSE;
        }
```

```
        if(form1.tel.value==""){                                  //判断电话是否为空
                    alert("请输入电话!");
                     form1.tel.select();
                      return FALSE;
         }
        if(!checktel(form1.tel.value)){                           //判断电话号码格式是否正确
                      alert("您输入的电话不正确!");
                      form1.tel.select();
                      return FALSE;
        }

              return TRUE;

}
</script>
```



图 6.23　验证用户注册信息是否合理

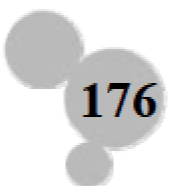
DIY：验证输入的用户名是否为汉字。提示：正则表达式格式为“/[^\u4E00-\u9FA5]/”。（20 分）（实例位置：光盘\mr\06\qjyy\02_diy）

6.6.3　情景应用 3——限制多行文本域输入的字符个数

视频讲解：光盘\mr\06\lx\限制多行文本域输入的字符个数.exe

实例位置：光盘\mr\06\qjyy\03

本例实现限制多行文本域输入的字符个数，关键是应用 onKeyDown 事件和 onKeyUp 事件，调用自定义的 JavaScript 函数 StatByte()，来限制输入的最多字符数并计算已用字节数和剩余字节数。运行结果如图 6.24 所示。



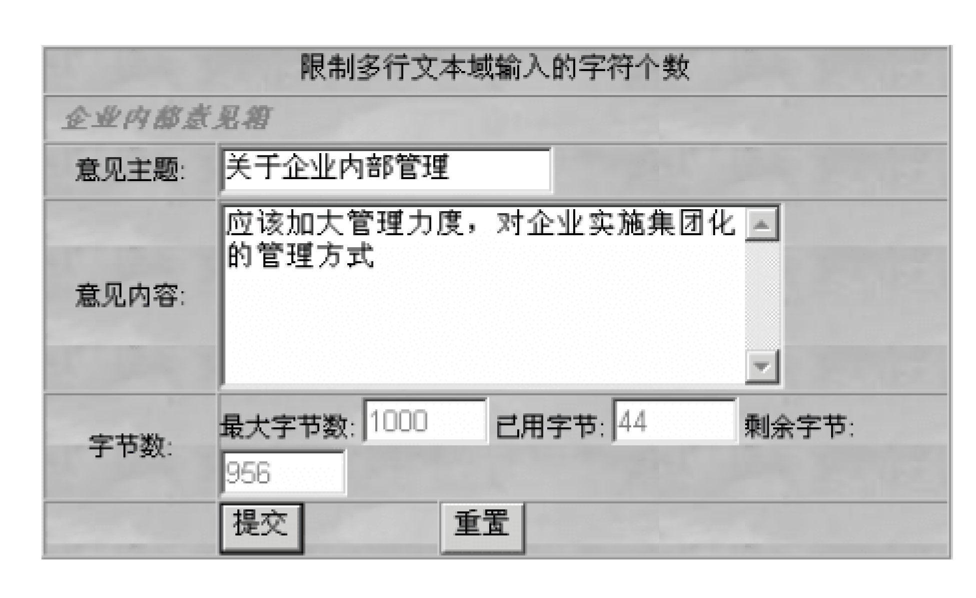


图 6.24　限制多行文本域输入的字符个数

具体步骤如下：

（1）创建一个企业内部意见箱的文本框，代码如下：

```
<textarea name="suggestions" cols="30" rows="5" class="wenbenkuang" id="suggestions">
     </textarea>
```

（2）添加显示字节数的文本框，包括最大字节数、已用字节数和剩余字节，注意此时应该将这 3 个文本框设置为停用状态，不允许用户修改。代码如下：

```
最大字节数:
  <input name="maxs" type="text" disabled class="noborder" id="maxs"   value="1000" size="4">
已用字节:
     <input name="used" type="text" disabled class="noborder" id="used"   value="0" size="4">
     剩余字节:
     <input name="remain" type="text" disabled class="noborder" id="remain" value="1000" size="4">
```

（3）编写自定义的 JavaScript 函数 StatByte()，用于判断用户输入的字符数是否大于允许输入的字符数，并计算已用字节数与剩余字节数。代码如下：

```
<SCRIPT language=JavaScript>
var LastCount =0;
function StatByte(Message,maxs,Used,Remain){ //字节统计
 var ByteCount = 0;
 var StrValue   = Message.value;
 var StrLength = Message.value.length;
 var maxsValue   = maxs.value;
if(LastCount != StrLength) {     //在此判断，减少循环次数
     for (i=0;i<StrLength;i++){
          ByteCount   = (StrValue.charCodeAt(i)<=256) ? ByteCount + 1 : ByteCount + 2;
       if (ByteCount>maxsValue) {
          Message.value = StrValue.substring(0,i);
          alert("留言内容最多不能超过 " +maxsValue+ " 个字节！\n 注意：一个汉字为两字节。");
          ByteCount = maxsValue;
          break;
       }
     }
   Used.value = ByteCount;
   Remain.value = maxsValue - ByteCount;
   LastCount = StrLength;
 }
}
</SCRIPT>
```

（4）修改“意见内容”文本域的代码，在 onKeyDown 事件和 onKeyUp 事件中调用自定义的 JavaScript

函数 StatByte()，限制输入的最多字符数并计算已用字节数和剩余字节数。代码如下：

```
<textarea name="suggestions" cols="30" rows="5" class="wenbenkuang" id="suggestions"
    onKeyDown="StatByte(this.form.suggestions,this.form.maxs,this.form.used,this.form.remain);"
onKeyUp="StatByte(this.form.suggestions,this.form.maxs,this.form.used,this.form.remain);">
    </textarea>>
```

DIY：通过表单中的文本域显示数据库中的信息。提示：设置文本域的初始值。（20 分）（实例位置：光盘\mr\06\qjyy\03_diy）

6.6.4 情景应用 4——设置文本框的只读属性

视频讲解：光盘\mr\06\lx\设置文本框的只读属性.exe

实例位置：光盘\mr\06\qjyy\04

本例实现将文本框设置为只读，运行结果如图 6.25 所示。

将文本框设置为只读属性有两种方法，一种是应用文本框的 readonly 属性实现，另一种是应用文本框的 disabled 属性实现。

图 6.25 设置文本框的只读属性

说明：在使用 readonly 和 disabled 属性时，只需在<input>标记中加入属性名称即可，例如：

```
<input type="text" readonly>
```

具体步骤如下：

（1）创建 conn.php 文件，连接数据库，代码如下：

```
<?php
$conn=mysql_connect('localhost','root','111');
mysql_select_db("db_database06",$conn);
mysql_query("set names utf8");
?>
```

（2）创建一个表格，在表格中插入文本框，在文本框中输出从数据库中读取的数据。设置文本框的相关属性值，在“商品编号”文本框的 id 中使用 disabled 属性来设置文本框的只读属性，在“库存数量”文本框的 count 中使用 readonly 属性来设置文本框的只读属性。关键代码如下：

```
<?php
include("conn.php");
$query=mysql_query("select * from tb_stock");
$myrow=mysql_fetch_array($query);
?>
<form name="form1" method="post" action="">
<input name="commodity_id" type="text" id="commodity_id" value="<?php echo $myrow['id'];?>" disabled
size="22" maxlength="50">
<input name="commodity_name" type="text" id="commodity_name" value="<?php echo $myrow['name'];?>"
size="22" maxlength="50">
<input name="stock_count" type="text" id="stock_count" value="<?php echo $myrow['count'];?>" readonly
onClick="alert('此文本框为只读属性,库存数量不能更改!')" size="22" maxlength="50">
<input name="price" type="text"class="text" id="price" value="<?php echo $myrow['price'];?>" size="10"
maxlength="50">
<input name="stock_money" type="text" id="stock_money" value="<?php echo $myrow['money'];?>" size="10"
maxlength="50">
  </form>
```

DIY：根据上面的实例，当用户输入库存数量和单价后，自动计算库存金额。提示：在单价文本框中设置其 onBlur 事件和按 Enter 键时执行的操作是调用 stock()函数。（20 分）（实例位置：光盘\mr\06\qjyy\04_diy）

6.6.5　情景应用 5——对超链接传递的数据进行编码

视频讲解：光盘\mr\06\lx\对超链接传递的数据进行编码.exe

实例位置：光盘\mr\06\qjyy\05

本例实现对超链接传递的数据进行编码，单击图片，通过超链接传递图片名称到指定文件页，应用 urlencode()函数对图片的名称进行编码，显示在 IE 地址栏中的字符串是编码后的字符串，运行结果如图 6.26 所示。

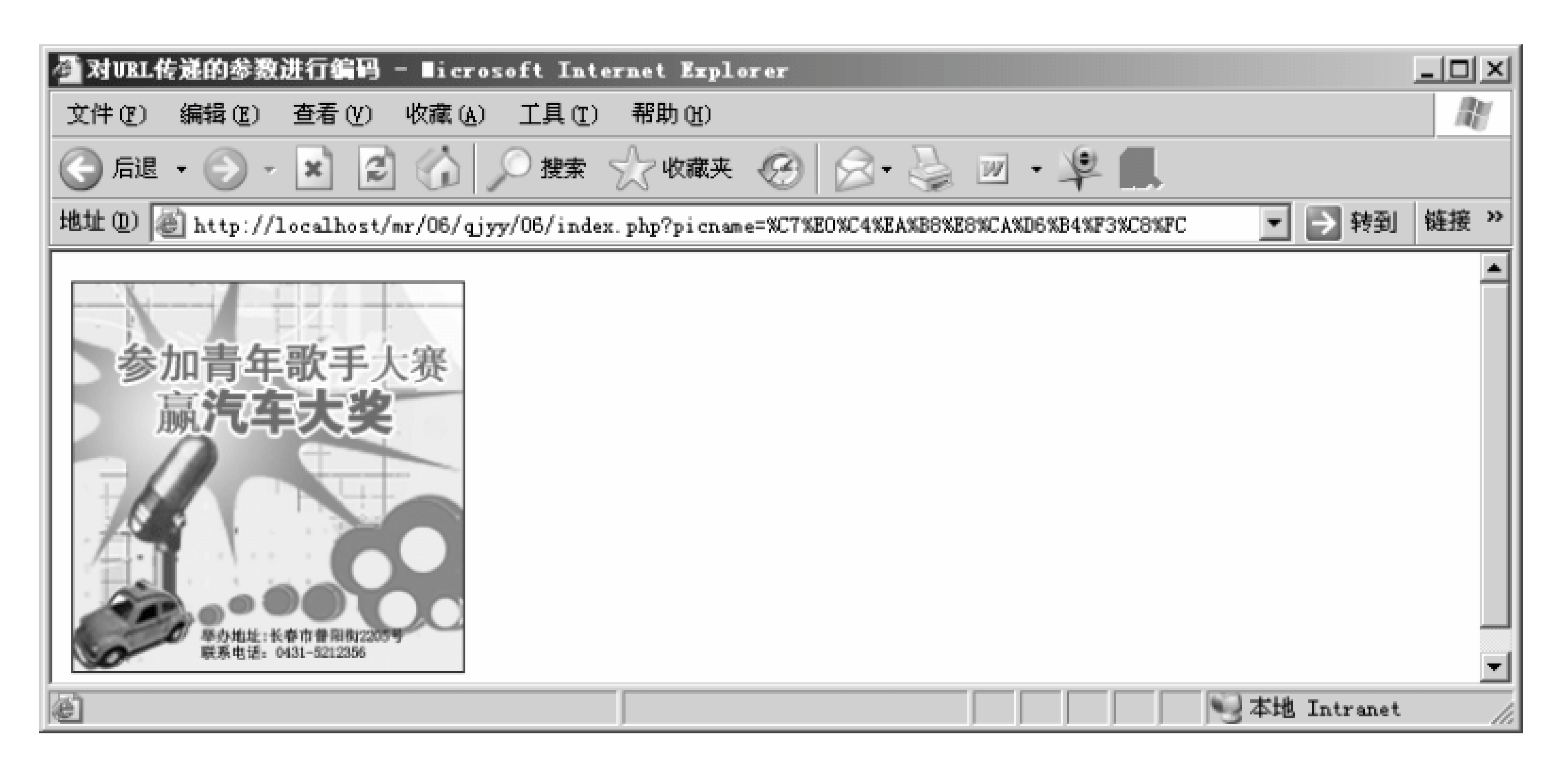


图 6.26　对超链接传递的数据进行编码

具体代码如下：

```
<a href="index.php?picname=<?php echo urlencode("青年歌手大赛");?>"><img src="images/music.jpg" width="200" height="200" border="1"></a>
```

DIY：对上例中编码后的超链接进行解码。提示：使用 unlencode()函数可以对获取后的字符串进行解码。（20 分）（实例位置：光盘\mr\06\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

6.7　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．在 JavaScript 脚本中，（　　）写法表示文本框（name）中的值。

A．$_POST['name']　　B．$_GET['name']　　C．var name　　D．form1.name.value

2．在 JavaScript 脚本中，弹出对话框，需使用 window 对象的（　　）方法。

A．echo　　B．window.location.href　　C．alert　　D．print

3．通过（　　）属性设置表单提交跳转页。

A．action　　B．method　　C．name　　D．value

4．对超链接传递的数据进行编码，使用（　　）函数。

A．action　　B．alert　　C．urldecode()　　D．urlencode()

5．通过超链接传递数据时，设置单击超链接时打开新的窗口，使用 target 属性的（　　）内置窗口名称。

A．_self　　B．_blank　　C．_parent　　D．_top

二、填空题（每题 10 分，5 道题）

1．函数参数传递方式分为（　　　　）和（　　　　）两种。

2．鼠标单击按钮时调用自定义函数使用（　　　　）事件。

3．通过超链接传递数据，使用（　　　　）方法接收。

4．自定义函数通常由 function 关键字、（　　　　）、（　　　　）、（　　　　）和（　　　　）5 部分组成。

5．在自定义函数中定义变量使用（　　　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

6.8 行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）在表单中添加一个隐藏域，并为其赋值，然后在另一页面中获取隐藏域的值。	
		（2）在表中添加一个下拉列表，并添加列表值，然后在另一页面中获取下拉列表的值。	
		（3）在表单中添加一个文本框，并设置为安全的密码域。	
		（4）在表单中添加一个链接，在另一页面中获取链接传递的值。	

续表

序号	内　　容	行 动 指 南
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

6.9　成功可以复制——软件业的华人教父王嘉廉

成就：全球第四大软件公司冠群电脑公司（CA）的创始人与领导者，王嘉廉被视为全球华人唯一能与比尔·盖茨比肩的传奇人物。

创始公司：冠群电脑公司

财富巅峰：资产范围 10 亿以上

出生日期：1944 年

最高学历：纽约皇后学院数学专业科学学士学位

超级语录：我们并不比其他公司的人更聪明，但不同的是，我们节省了许多相互指责的时间。从错误中学到教训，不断向上成长。

成功历程

王嘉廉（Charles Wang）1944 年出生于上海法租界内，父亲曾是最高法院的法官。1952 年，8 岁的王嘉廉随父母移民美国，先是进入布鲁克林科技高中就读，后在市立大学皇后学院毕业。那时生活异常艰辛，王嘉廉常做一些被人瞧不起的活，经常连 32 美分一顿的午饭都吃不上。许多年后，回忆起早年的痛苦，他总是轻描淡写地带过，只是常说自己比比尔·盖茨经历的磨难多。“我知道饥饿的滋味，而他却不知道。”

虽然与盖茨将近 1000 亿美元的个人财富相比，王嘉廉的 10 多亿个人资产不过是盖茨的一个零头，但是人们还是常常把他们放在一起，他们都是自己公司的中心和灵魂，都是凭借铁腕式的战略攻城略地，成为软件业头两号厂商。微软在 PC 领域一统天下，CA 的企业管理软件在大型机和客户机/服务器领域也是领袖群伦。

王嘉廉在皇后大学毕业后就结束了学习生涯。毕业后，王嘉廉进入哥伦比亚大学电子研究实验室当了一位程序员新手，结果一下子就爱上了编程。他在好几家公司从事软件工作，其中包括标准数据公司。在

那里，他和大学的伙伴 Russ Artzt 一并出售用于 IBM 大型机的系统软件。两人经常拜访客户，倾听企业信息系统管理员反映的各种问题，从而看到了商机，萌生了创业的念头。1976 年，他终于可以将想法付诸于实践。一家瑞士的 CA 国际公司，正在寻求一家美国公司，代理销售 CA-EARL 大型机软件。后来 CA 找到了标准数据，双方成立了一个合资公司。几个月后，标准数据准备放弃软件业务。王嘉廉、Artzt 和另外两位朋友 Sedino 和 Habermass 就成立了 CA 国际公司，作为瑞士公司的子公司。在曼哈顿的麦迪逊大街的一间办公室里，他们开始推销 CA-SORT。当时条件十分艰难，他们是用服务换回了上机室和办公室。Sedino 为房东担任招待员，Artzt 和 Habermass 负责产品开发，王嘉廉则是销售和市场部的光杆司令，负责销售业务。开始没有任何业绩，王嘉廉就为大楼的休息厅铺地毯、在走廊中挂镜子，以抵消房租。这家位于纽约的软件公司从一开始可以说一文不名，但 1980 年 4 月，他们却以 280 万美元买下了瑞士的母公司，从此 CA 完全属于他们。也就是从这时开始，CA 实施了一系列的并购举措，以实现高速增长。



CA 产品

深度评价

尽管人们常说计算机行业是年轻人拼搏的战场，但也有宝刀不老者，55 岁的王嘉廉就是例证。他领导的 CA 在国际软件业是知名的公司，他本人也因为华人却在白人领域中闯出一番天地而更具传奇色彩。王嘉廉的勤奋、刻苦、善于把握时机、勇往直前的精神，永远值得每一个渴望成功的人学习。

第 7 堂课

MySQL 数据库技术

（视频讲解：155 分钟）

数据库作为程序中数据的主要载体，在整个项目中扮演着重要的角色。PHP 自身可以与大多数数据库进行连接，但 MySQL 数据库是开源界所公认的与 PHP 结合最好的数据库，其具有安全、跨平台、体积小和高效等特点，可谓 PHP 的“黄金搭档”。在本堂课中将对 MySQL 数据库的基础知识进行系统的讲解，为在第 8 堂课中实现 PHP 与 MySQL 数据库的完美结合打下坚实的基础。

学习摘要：

- **MySQL 概述**
- **MySQL 服务器的启动和关闭**
- **操作 MySQL 数据库**
- **操作 MySQL 数据表**
- **操作 MySQL 数据**
- **MySQL 数据类型**
- **phpMyAdmin 管理 MySQL 数据库**

7.1　MySQL 概述

学习编程语言，至少要掌握一种数据库，学习 PHP 语言，则非常有必要掌握 MySQL。虽然现在 PHP 对数据库的支持越来越多，如 Access、SQL Server、Oracle、DB2 等，但是 LAMP 的开发模式中，MySQL 仍然牢牢占据一席之地。

7.1.1　MySQL 的特点

- ☑ MySQL 是一个关系数据库管理系统，把数据存储在表格中，使用标准的结构化查询语言——SQL 访问数据库。
- ☑ MySQL 是完全免费的，在网上可以任意下载，并且可以查看到它的源文件，进行必要的修改。
- ☑ MySQL 服务器的功能齐全，运行的速度极快，十分可靠，有很好的安全性。
- ☑ MySQL 服务器在客户、服务器或嵌入系统中使用，是一个客户机/服务器系统，能够支持多线程、多个不同的客户程序和管理工具。

7.1.2　SQL 和 MySQL

SQL（Structured Query Language，结构化查询语言）与其说是一门语言，倒不如说是一种标准，数据库系统的工业标准。大多数的 RDBMS 开发商的 SQL 都基于这个标准，虽然在有些地方并不是完全相同，但这并不妨碍对 SQL 的学习和使用。

下面给出 SQL 标准的关键字及其功能，如表 7.1 所示。

表 7.1　SQL 标准的关键字及其功能

功 能 类 型	SQL 关键字	功　　能
数据查询语言	Select	从一个或多个表中查询数据
数据定义语言	Create/Alter/Drop table Create/Alter/Drop index	创建/修改/删除表 创建/修改/删除索引
数据操纵语言	Insert Delete Update	向表中插入新数据 删除表中的数据 更新表中现有的数据
数据控制语言	Grant Revoke	为用户赋予特权 收回用户的特权

在 MySQL 中，不仅支持 SQL 标准，而且还对其进行了扩展，使得它能够支持更为强大的功能。下面给出 MySQL 支持的 SQL 关键字，如表 7.2 所示。

表 7.2　MySQL 支持的 SQL 关键字

SQL 关键字	功　　能
Create/Drop database/Use	创建、删除和选择数据库
Create/Alter/Drop table Create/Alter/Drop index	创建、更改和删除表/索引



续表

SQL 关键字	功　能
Select	查询表中的信息
Describe、Explain、Show	取数据库、表和查询的有关信息
Delete、Insert、Update、Load data、Optimize table、Replace	修改表中的信息
Flush、Grant、Kill、Revoke	管理语句
Create/Drop function、Lock/Unlock tables、Set	其他语句

在 MySQL 中，可以直接使用 SQL 语句，这些语句几乎可以不加修改地嵌入到 PHP 语言中去。另外，MySQL 还允许在 SQL 语句中使用注释，有 3 种编写注释的方式：

☑　以“#”号开头直到行尾的所有内容都是注释。

☑　以“-- ”号开头直到行尾的所有内容都是注释，注意在“--”后面还有一个空格。

☑　以“/*”开始，以“*/”结束的所有内容都是注释，可以对多行进行注释。

7.2　MySQL 服务器的启动和关闭

通过系统服务器、命令提示符（DOS）都可以启动和停止 MySQL，操作非常简单。但通常情况下，不要暂停或停止 MySQL 服务器，否则数据库将无法使用。

7.2.1　启动 MySQL 服务器

启动 MySQL 服务器的方法有两种，即分别通过系统服务器和命令提示符（DOS）启动。

1．通过系统服务器启动 MySQL

选择“开始”/“设置”/“控制面板”/“管理工具”/“服务”命令，打开“服务”面板，从“名称”列中找到 MySQL 服务，单击鼠标右键，在弹出的快捷菜单中选择“所有任务”/“启动”命令，如图 7.1 所示。



图 7.1　通过系统服务启动 MySQL

2．在命令提示符下启动 MySQL

选择“开始”/“运行”命令，在弹出的对话框中输入“cmd”，进入 DOS 窗口，在命令提示符下输入如下指令：

```
\> net start MySQL
```

按 Enter 键就会看到启动信息，如图 7.2 所示。



图 7.2　在命令提示符下启动 MySQL

7.2.2　连接 MySQL 服务器

MySQL 服务器启动后，接下来就是连接服务器。连接 MySQL 服务器也有两种方法：一种是进入 DOS 窗口，通过命令来连接；另一种是使用 MySQL 数据库系统函数连接。有关通过 MySQL 数据库系统函数连接数据库的方法将在第 8 堂课中进行详细讲解，这里只介绍在命令提示符（DOS）中操作数据库的方法。

选择“开始”/“运行”命令，在弹出的对话框中输入“cmd”，进入 DOS 窗口。在命令提示符下输入如下指令：

```
\> MySQL –uroot –h127.0.0.1 –ppassword
```

其中，-u 后输入的是用户名 root；-h 后输入的是 MySQL 数据库服务器地址；-p 后输入的是密码。

输入完命令语句后，按 Enter 键就进入到 MySQL 数据库中，如图 7.3 所示。

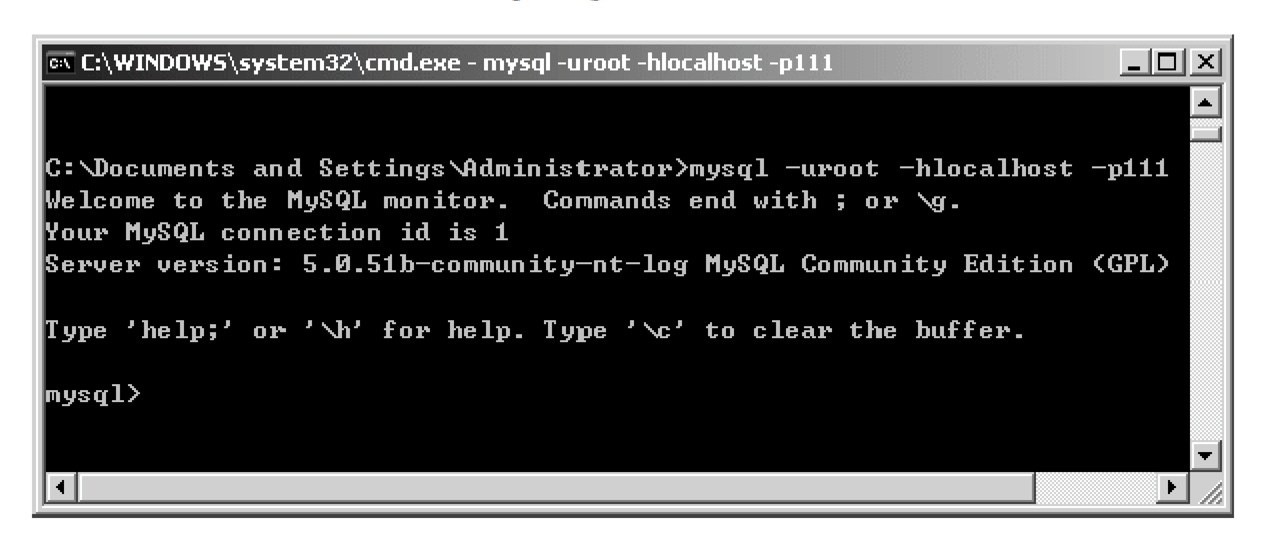


图 7.3　连接 MySQL

说明： 在输入的用户名“-uroot”与“-h127.0.0.1”之间必须有一个空格，同样在“-h127.0.0.1”与“-ppassword”之间也必须有一个空格，同时在这个命令的结束之处不必书写“;”。

技巧： 为了保护 MySQL 数据库的密码，可以采用如图 7.4 所示的密码输入方式。如果密码在-p 后直接给出，那么密码就以明文显示，如 MySQL -u root -h 127.0.0.1 -p 111。按 Enter 键后再输入密码将以加密的方式显示，然后按 Enter 键即可成功连接 MySQL 服务器。

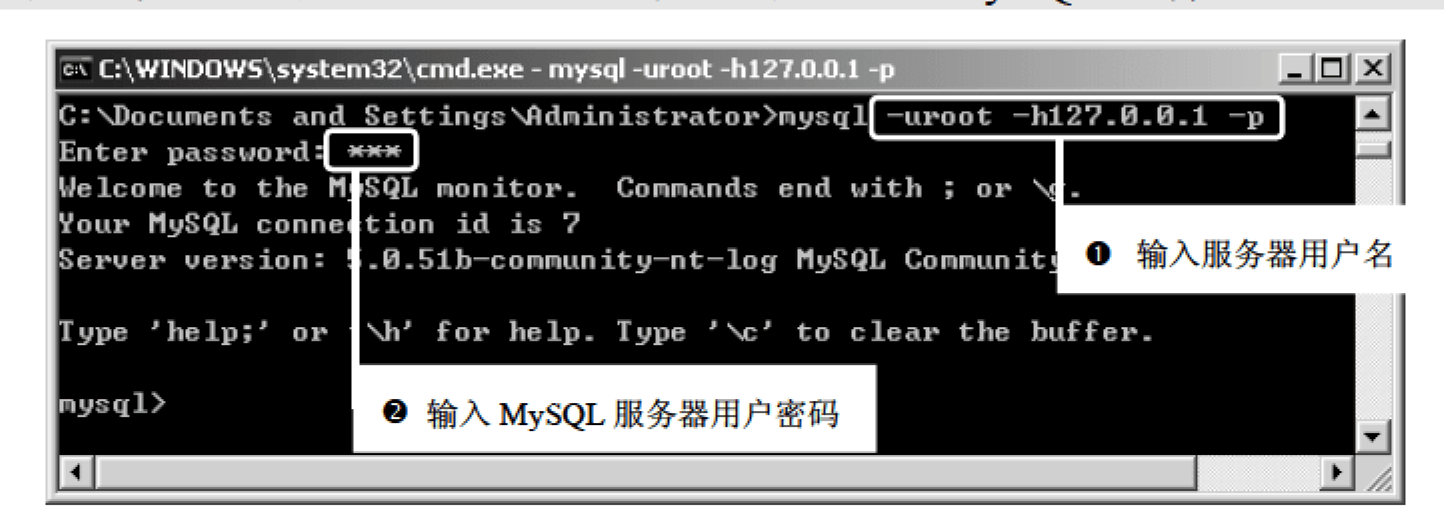


图 7.4　以隐藏密码方式连接服务器



7.2.3　关闭 MySQL 服务器

关闭 MySQL 服务器的方法也有通过系统服务和命令提示符（DOS）启动两种。

1．通过系统服务器关闭 MySQL

选择“开始”/“设置”/“控制面板”/“管理工具”/“服务”命令，打开“服务”面板，从“名称”列表中找到 MySQL 服务，单击鼠标右键，在弹出的快捷菜单中选择“所有任务”/“停止”命令，如图 7.5 所示。

2．在命令提示符下关闭 MySQL

选择“开始”/“运行”命令，在弹出的对话框中输入“cmd”，进入 DOS 窗口，在命令提示符下输入如下指令：

```
\> net stop MySQL
```

按 Enter 键后可看到提示服务停止的信息，如图 7.6 所示。



图 7.5　停止 MySQL 服务器



图 7.6　在命令提示符中关闭 MySQL 服务器

7.3　操作 MySQL 数据库

MySQL 数据库的基本操作可以分为创建、选择和删除 3 种。

7.3.1　创建新数据库

在 MySQL 中，应用 CREATE DATABASE 语句创建数据库。其语法格式如下：

```
CREATE DATABASE db_name ;
```

其中，db_name 是要创建的数据库名称，该名称必须是合法的，对于数据库的命名，有如下规则：

☑　不能与其他数据库重名。
☑　名称由任意字母、阿拉伯数字、下划线（_）或者“$”组成，可以使用上述的任意字符开头，但不能使用单独的数字，那样会造成它与数值相混淆。
☑　名称最长可由 64 个字符组成（还包括表、列和索引的命名），而别名最多可长达 256 个字符。
☑　不能使用 MySQL 关键字作为数据库、表名。

下面通过 CREATE DATABASE 语句创建一个名称为 db_database14 的数据库，运行结果如图 7.7 所示。

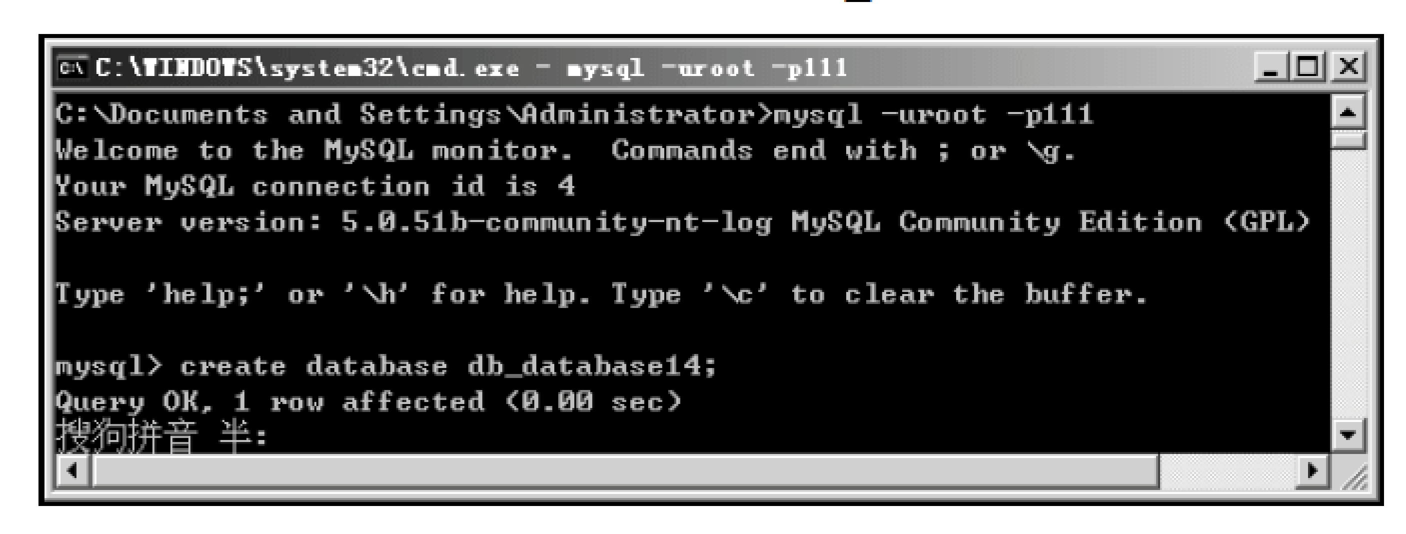


图 7.7　创建数据库

说明：在创建数据库时，首先连接 MySQL 服务器，用户名是 root，密码是 111，然后编写“create database db_database14;” SQL 语句，数据库创建成功。

7.3.2　选择指定数据库

USE 语句用于选择一个数据库，使其成为当前默认数据库。其语法如下：

```
USE db_name ;
```

例如，选择名称为 db_database14 的数据库，如图 7.8 所示。

```
C:\WINDOWS\system32\cmd.exe - mysql -...
mysql> create database db_database14;
Query OK, 1 row affected (0.00 sec)

mysql> use db_database14;
Database changed
mysql> _
```

图 7.8　选择数据库

如图 7.8 所示，已经进入到 db_database14 数据库中。

7.3.3　删除指定数据库

删除数据库使用的是 DROP DATABASE 语句，语法如下：

```
DROP DATABASE db_name ;
```

例如，通过 DROP DATABASE 语句删除名称为 db_database14 的数据库，如图 7.9 所示。

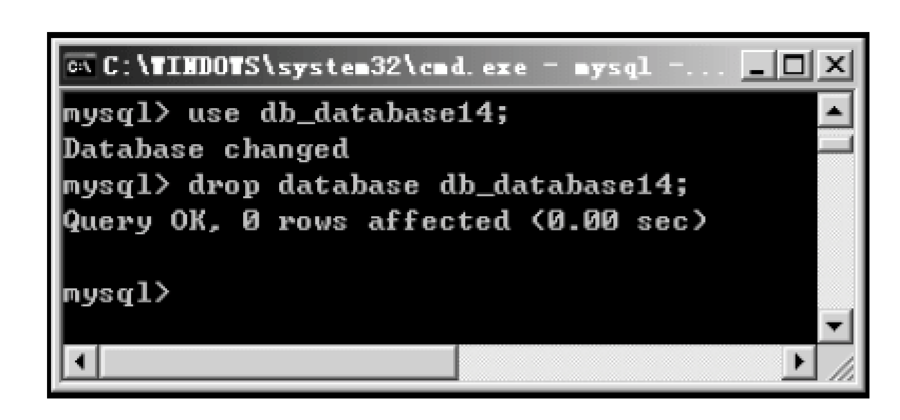
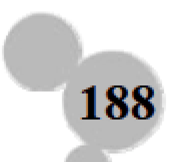


图 7.9　删除数据库

说明：对于删除数据库的操作，应该谨慎使用，一旦执行这项操作，数据库的所有结构和数据都会被删除，没有恢复的可能，除非数据库有备份。



7.4　操作 MySQL 数据表

MySQL 数据表的基本操作包括创建、查看、修改、重命名和删除。

7.4.1　创建一个表

创建数据表使用 CREATE TABLE 语句，语法如下：

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] 数据表名
[(create_definition,…)][table_options] [select_statement]
```

CREATE TABLE 语句的参数说明如表 7.3 所示。

表 7.3　CREATE TABLE 语句的参数说明

关　键　字	说　　明
TEMPORARY	如果使用该关键字，表示创建一个临时表
IF NOT EXISTS	该关键字用于避免表存在时 MySQL 报告的错误
create_definition	这是表的列属性部分。MySQL 要求在创建表时，表要至少包含一列
table_options	表的一些特性参数
select_statement	SELECT 语句描述部分，用它可以快速创建表

下面介绍列属性 create_definition 部分，每一列定义的具体格式如下：

```
col_name   type [NOT NULL | NULL] [DEFAULT default_value] [AUTO_INCREMENT]
              [PRIMARY KEY ] [reference_definition]
```

属性 create_definition 的参数说明如表 7.4 所示。

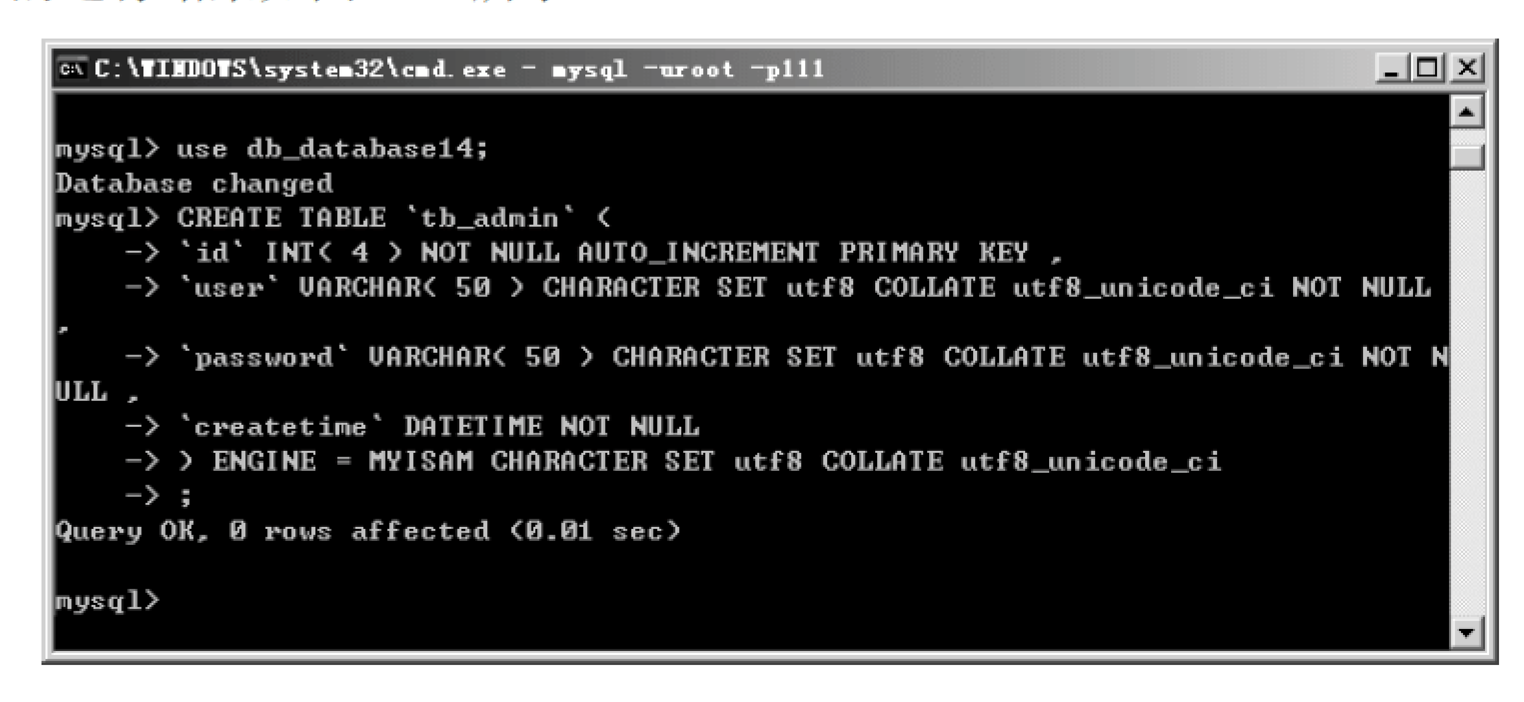
表 7.4　属性 create_definition 的参数说明

参　　数	说　　明
col_name	字段名
type	字段类型
NOT NULL | NULL	指出该列是否允许是空值，系统一般默认允许为空值，所以当不允许为空值时，必须使用 NOT NULL
DEFAULT default_value	表示默认值
AUTO_INCREMENT	表示是否是自动编号，每个表只能有一个 AUTO_INCREMENT 列，并且必须被索引
PRIMARY KEY	表示是否为主键。一个表只能有一个 PRIMARY KEY。如果表中没有 PRIMARY KEY，而某些应用程序又需要 PRIMARY KEY，MySQL 将返回第一个没有任何 NULL 列的 UNIQUE 键，作为 PRIMARY KEY
reference_definition	为字段添加注释

例如，使用 CREATE TABLE 语句在 MySQL 数据库 db_db_database14 中创建一个名为 tb_admin 的简单的数据表，该表包括 id、user、password 和 createtime 4 个字段。创建的 SQL 语句如下：

```
CREATE TABLE `tb_admin` (
`id` INT( 4 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
`user` VARCHAR( 50 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,
`password` VARCHAR( 50 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,
`createtime` DATETIME NOT NULL ) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

在命令模式下的运行结果如图 7.10 所示。



```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -p111
mysql> use db_database14;
Database changed
mysql> CREATE TABLE `tb_admin` (
    -> `id` INT( 4 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
    -> `user` VARCHAR( 50 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL
,
    -> `password` VARCHAR( 50 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT N
ULL ,
    -> `createtime` DATETIME NOT NULL
    -> ) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_unicode_ci
    -> ;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

图 7.10　创建 MySQL 数据库

技巧： 在输入 SQL 语句时，可以一行全部输出，也可以每个字段都换行输出，这里建议用换行输出，这样看上去美观、易懂，在语句出现错误时更容易查找。

7.4.2　查看数据表结构

对于一个创建成功的数据表，可以使用 SHOW COLUMNS 语句或 DESCRIBE 语句查看该数据表的结构。

1．SHOW COLUMNS 语句

SHOW COLUMNS 语句用于查看一个指定的数据表，语法如下：

```
SHOW  [FULL] COLUMNS  FROM 数据表名 [FROM 数据库名];
```

或写成

```
SHOW  [FULL] COLUMNS  FROM 数据表名.数据库名;
```

例如，使用 SHOW COLUMNS 语句查看数据表 tb_admin 的结构，如图 7.11 所示。

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -p111
mysql> show columns from tb_admin ;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| id         | int(4)      | NO   | PRI | NULL    | auto_increment |
| user       | varchar(50) | NO   |     | NULL    |                |
| password   | varchar(50) | NO   |     | NULL    |                |
| createtime | datetime    | NO   |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

图 7.11　查看表结构

2．DESCRIBE 语句

DESCRIBE 语句的语法如下：

```
DESCRIBE 数据表名;
```

其中，DESCRIBE 可以简写成 DESC。在查看表结构时，也可以只列出某一列的信息。其语法格式如下：

```
DESCRIBE 数据表名 列名;
```

例如，使用 DESCRIBE 语句的简写形式查看数据表 tb_admin 中的某一列信息，如图 7.12 所示。

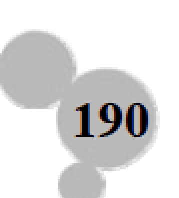




图 7.12　查看表的某一列信息

7.4.3　修改数据表结构

修改表结构使用 ALTER TABLE 语句。修改表结构的操作包括增加或者删除字段、修改字段名称或者字段类型、设置取消主键外键、设置取消索引以及修改表的注释等。其语法如下：

```
ALTER[IGNORE] TABLE 数据表名 alter_spec[,alter_spec]…
```

其中，alter_spec 子句用来定义要修改的内容，语法如下：

```
alter_specification:
    ADD [COLUMN] create_definition [FIRST | AFTER column_name ]          //添加新字段
  | ADD INDEX [index_name] (index_col_name,...)                         //添加索引名称
  | ADD PRIMARY KEY (index_col_name,...)                                //添加主键名称
  | ADD UNIQUE [index_name] (index_col_name,...)                        //添加唯一索引
  | ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}        //修改字段名称
  | CHANGE [COLUMN] old_col_name create_definition                      //修改字段类型
  | MODIFY [COLUMN] create_definition                                   //修改子句定义字段
  | DROP [COLUMN] col_name                                              //删除字段名称
  | DROP PRIMARY KEY                                                    //删除主键名称
  | DROP INDEX index_name                                               //删除索引名称
  | RENAME [AS] new_tbl_name                                            //更改表名
  | table_options
```

ALTER TABLE 语句允许指定多个动作，其动作间使用逗号分隔，每个动作表示对表的一个修改。

说明： 在使用 ALTER TABLE 语句修改数据表时，如果指定 IGNORE 参数，当出现重复的行时，则只执行一行，其他重复的行被删除。

例如，添加一个新的字段 email，类型为 varchar(50)、not null，将字段 user 的类型由 varchar(50)改为 varchar(80)，代码如下：

```
alter table tb_admin add email varchar(50) not null ,modify user varchar(80);
```

在命令模式下的运行情况如图 7.13 所示。

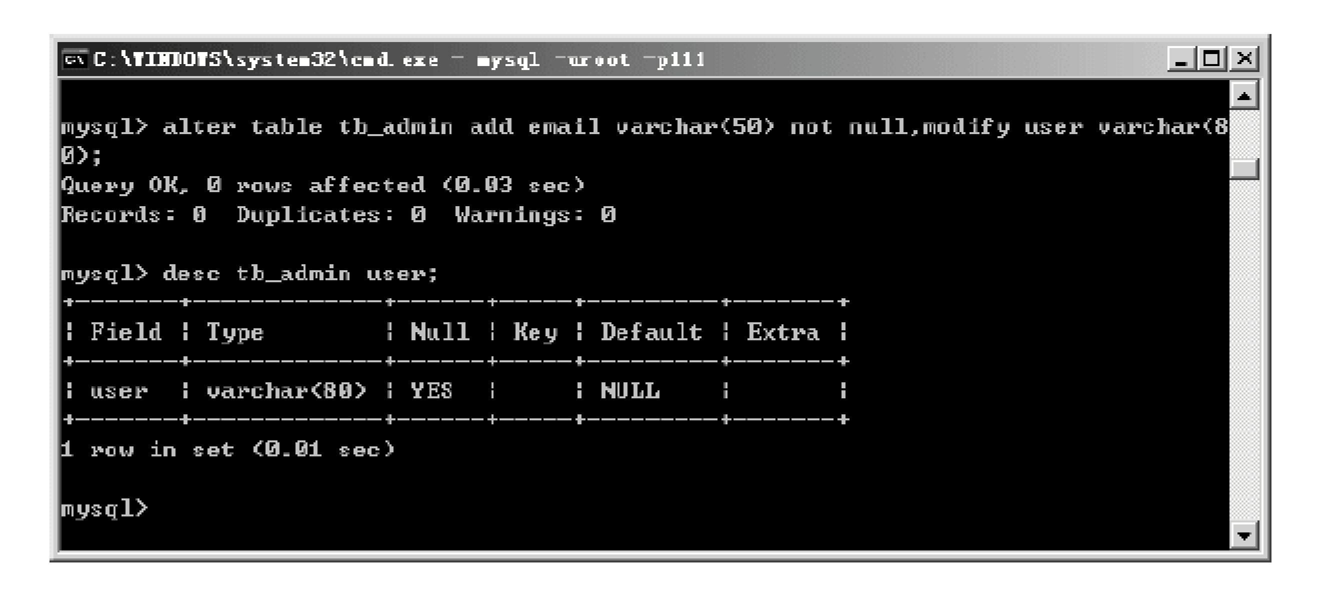


图 7.13　修改表结构

图 7.13 中只给出了修改 user 字段类型的结果，读者可以通过语句“MySQL> show tb_admin;”查看整个表的结构，以确认 email 字段是否添加成功。

说明： 通过 alter 修改表列，其前提是必须将表中的数据全部删除，然后才可以修改。

7.4.4　重命名数据表

重命名数据表使用 RENAME TABLE 语句，语法如下：

```
RENAME TABLE 数据表名 1 To 数据表名 2
```

注意： 使用 RENAME TABLE 语句可以同时对多个数据表进行重命名，多个表之间以逗号“,”分隔。

例如，对数据表 tb_admin 进行重命名，更名后的数据表为 tb_user，如图 7.14 所示。



图 7.14　对数据表进行更名

7.4.5　删除指定数据表

删除数据表使用 DROP TABLE 语句，语法如下：

```
DROP TABLE 数据表名;
```

例如，删除数据表 tb_user，如图 7.15 所示。



图 7.15　删除数据表

在删除数据表的过程中删除一个不存在的表将会产生错误，如果在删除语句中加入 if exists 关键字则不会出错。其语法如下：

```
drop table if exists 数据表名;
```

说明： 删除数据表的操作应该谨慎使用。一旦删除了数据表，那么表中的数据将会全部清除，没有备份则无法恢复。

技巧： 无论在执行 CREATE TABLE、ALTER TABLE 还是 DROP TABLE 中的任何操作时，首先必须要选择数据库，否则是无法对数据表进行操作的。



7.5　操作 MySQL 数据

数据库中包含数据表，而数据表中包含数据。在 MySQL 与 PHP 的结合应用中，真正被操作的是数据表中的数据，因此如何更好地操作和使用这些数据才是使用 MySQL 数据库的根本。

7.5.1　向数据表中添加数据

建立一个数据库，并在数据库中创建一个数据表，首先要向数据表中添加数据。这项操作可以通过 INSERT 语句来完成。

下面通过向 db_database07 数据库的 tb_mrbook 表中添加一条记录，介绍 INSERT 语句的 3 种语法。

（1）列出新添加数据的所有的值。其语法如下：

```
insert into table_name values (value1,value2, … )
```

（2）给出要赋值的列，然后再给出值。其语法如下：

```
insert into table_name (column_name,column_name2, … ) values (value1, value1, … )
```

（3）用 col_name = value 的形式给出列和值。其语法如下：

```
insert into table_name set column_name1 = value1,column_name2 = value2, …
```

技巧： 采用第一种语法格式的好处是输入的 SQL 语句短小，错误查找方便。不利的是，如果字段很多，不容易看到数据是属于哪一列的，不易匹配。

你问我答： 如何执行数据的批量添加？

数据的批量添加使用 LOAD DATA 和 MySQLIMPORT 语句实现。

LOAD DATA 语句，通过读取本地文件系统上的文件，可以将大量数据添加到数据库中。其语法如下：

```
load data local infile "filename.txt" into table table_name
```

其中，filename.txt 是当前目录中的数据文件的名称；table_name 是要被装载的数据表。

LOAD DATA 语句读取客户机当前目录中数据文件 filename.txt 的内容，将其内容装载到数据表 table_name 中。

MySQLIMPORT 语句使程序直接从文件读取批量数据。它相当于 LOAD DATA 语句的一个接口。其语法如下：

```
%MySQLimport -local table_name filename.txt ;
```

MySQLIMPORT 可以自动生成一个 LOAD DATA 语句，该语句把 filename.txt 文件中的数据装入 table_name 表中。MySQLIMPORT 根据文件名导出表名，即将文件名第一个圆点前的所有字符作为表名。例如，文件 class.txt 被装入 class 表中。

7.5.2　更新数据表中数据

修改数据使用 UPDATE 语句，语法如下：

```
update table_name
set column_name = new_value1,column_name2 = new_value2, …
where condition
```

其中 table_name 是更新的表名称；SET 子句指出要修改的列和它们给定的值；WHERE 子句是可选的，如果应用它将指定记录中哪行应该被更新，否则，所有的记录行都将被更新。

例如，下面将管理员信息表 tb_user 中用户名为 tsoft 的管理员密码 111 修改为 123456，如图 7.16 所示。

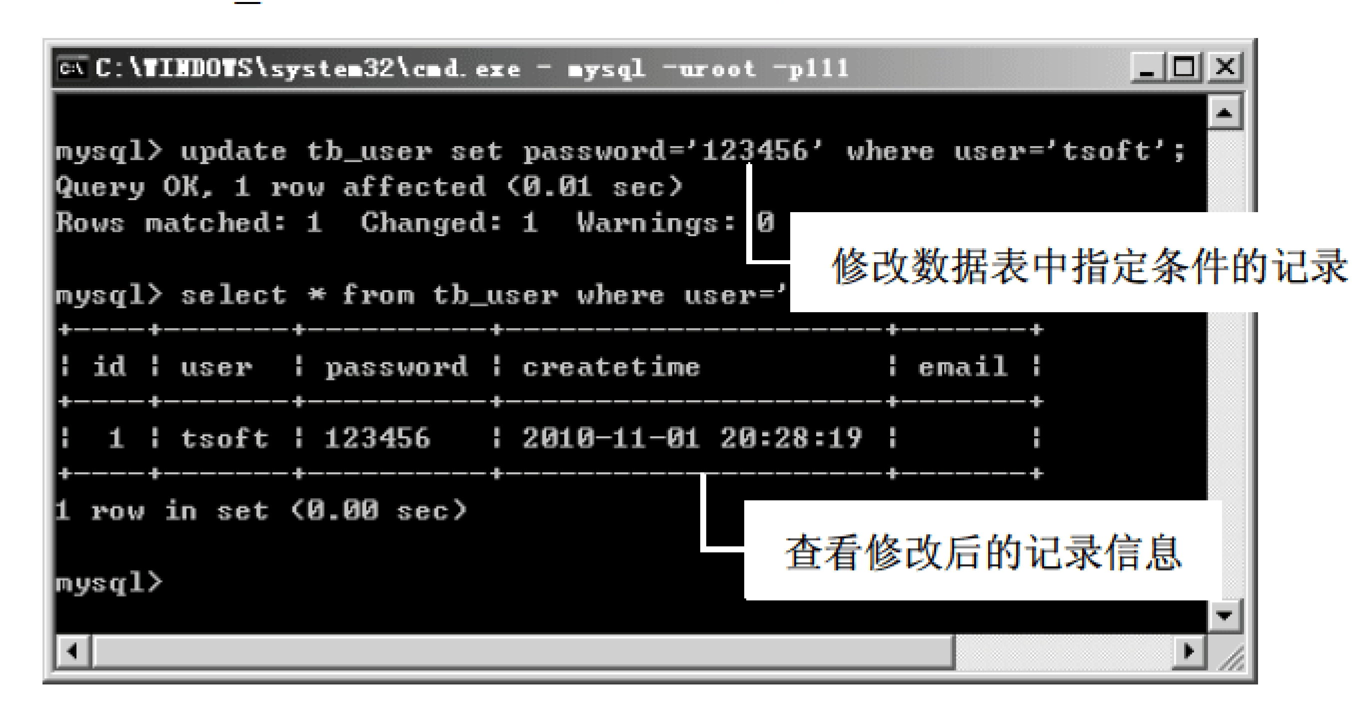


图 7.16　修改指定条件的记录

注意： 更新时一定要保证 where 子句的正确性，一旦 where 子句出错，将会破坏所有改变的数据。

7.5.3　删除数据表中数据

删除数据使用 DELETE 语句，语法如下：

```
delete from table_name
where condition
```

该语句在执行过程中删除 table_name 表中的记录，如果没有指定 WHERE 条件，将删除所有的记录；如果指定 WHERE 条件，将按照指定的条件进行删除。

例如，删除管理员数据表 tb_user 中用户名为 mr 的记录信息，如图 7.17 所示。



图 7.17　删除数据表中指定的记录

技巧：（1）在实际的应用中，在执行删除操作时，执行删除的条件一般应该为数据的 id，而不是具体某个字段值，这样可以避免一些不必要的错误发生。

（2）使用 DELETE 语句删除整个表的效率并不高，还可以使用 TRUNCATE 语句，它可以很快地删除表中所有的内容。

7.5.4　查询数据表中数据

创建数据库的目的不只是保存数据，更重要的是为了使用其中的数据。要从数据库中把数据查询出来，使用的是 SELECT 查询语句。SELECT 语句的语法如下：

```
select selection_list                    //要查询的内容，选择哪些列
from table_list                          //从什么表中查询，从何处选择行
```

```
where primary_constraint                          //查询时需要满足的条件，行必须满足的条件
group by grouping_columns                         //如何对结果进行分组
order by sorting_cloumns                          //如何对结果进行排序
having secondary_constraint                       //查询时满足的第二条件
limit count                                       //限定输出的查询结果
```

以上就是 SELECT 查询语句的语法，下面对它的参数进行详细的讲解。

1．selection_list

设置查询内容。如果要查询表中所有列，可以将其设置为“*”；如果要查询表中某一列或多列，则直接输入列名，并以“,”为分隔符。

例如，查询 tb_mrbook 数据表中所有列和查询 user 和 pass 列的代码如下：

```
select * from tb_mrbook;                          //查询数据表中所有数据
select user,pass from tb_mrbook;                  //查询数据表中 user 和 pass 列的数据
```

2．table_list

指定查询的数据表。既可以从一个数据表中查询，也可以从多个数据表中进行查询，多个数据表之间使用“,”分隔，并且通过 where 子句使用连接运算来确定表之间的联系。

例如，从 tb_mrbook 和 tb_bookinfo 数据表中查询 bookname='PHP 入门与精通'的作者和价格。代码如下：

```
select tb_mrbook.id,tb_mrbook.bookname,
    -> author,price from tb_mrbook,tb_bookinfo
    -> where tb_mrbook.bookname = tb_bookinfo.bookname and
    -> tb_bookinfo.bookname = 'PHP 入门与精通';
```

在上面的 SQL 语句中，因为两个表都有 id 字段和 bookname 字段，为了告诉服务器要显示哪个表中的字段信息，要加上前缀。其语法如下：

```
表名.字段名
```

tb_mrbook.bookname = tb_bookinfo.bookname 将表 tb_mrbook 和 tb_bookinfo 连接起来，叫做等同连接；如果不使用 tb_mrbook.bookname = tb_bookinfo.bookname，那么产生的结果将是两个表的笛卡儿积，叫做全连接。

3．where 条件语句

在使用查询语句时，如要从很多的记录中查询出想要的记录，就需要一个查询的条件。只有设定了查询的条件，查询才有实际的意义。设定查询条件应用的是 where 子句。

where 子句的功能非常强大，通过它可以实现很多复杂的条件查询。在使用 where 子句时，需要使用一些比较运算符，常用的比较运算符如表 7.5 所示。

表 7.5　常用的 where 子句比较运算符

运　算　符	名　　称	示　　例	运　算　符	名　　称	示　　例
=	等于	id=5	is not null	n/a	id is not null
>	大于	id>5	between	n/a	id between1 and 15
<	小于	id<5	in	n/a	id in (3,4,5)
=>	大于等于	id=>5	not in	n/a	name not in (shi,li)
<=	小于等于	id<=5	like	模式匹配	name like (‘shi%’)
!=或<>	不等于	id!=5	not like	模式匹配	name not like (‘shi%’)
Is null	n/a	id is null	regexp	常规表达式	name 正则表达式

表 7.5 中列举的示例中的 id 是记录的编号，name 是表中的用户名。

例如，应用 where 子句查询 tb_mrbook 表，条件是 type（类别）为 PHP 的所有图书。代码如下：

```
select * from tb_mrbook where type = 'php';
```

4．GROUP BY 子句

通过 GROUP BY 子句可以将数据划分到不同的组中，实现对记录进行分组查询。在查询时，所查询的列必须包含在分组的列中，目的是使查询到的数据没有矛盾。在与 AVG()或 SUM()函数一起使用时，GROUP BY 子句能发挥最大作用。

例如，查询 tb_mrbook 表，按照 type 进行分组，求每类图书的平均价格。代码如下：

```
select bookname,avg(price),type from tb_mrbook group by type;
```

5．DISTINCT 关键字

使用 DISTINCT 关键字可以去除结果中重复的行。

例如，查询 tb_mrbook 表，并在结果中去掉类型字段 type 中的重复数据。代码如下：

```
select distinct type from tb_mrbook;
```

6．ORDER BY 子句

使用 ORDER BY 子句可以对查询的结果进行升序和降序（DESC）排列，在默认情况下，ORDER BY 按升序输出结果。如果要按降序排列可以使用 DESC 来实现。

对含有 null 值的列进行排序时，如果是按升序排列，null 值将出现在最前面；如果是按降序排列，null 值将出现在最后。

例如，查询 tb_mrbook 表中的所有信息，按照 id 进行降序排列，并且只显示 3 条记录。代码如下：

```
select * from tb_mrbook order by id desc limit 3;
```

7．LIKE 运算符

LIKE 属于较常用的比较运算符，通过它可以实现模糊查询。它有“%”和下划线“_”两种通配符。“%”可以匹配一个或多个字符，而“_”只匹配一个字符。

例如，查找所有第二个字母是“h”的图书，代码如下：

```
select * from tb_mrbook where bookname like('_h%');
```

说明：“p”和“入”都算作一个字符，这一点上英文字母和中文没有什么区别。

8．CONCAT 函数

使用 CONCAT 函数可以联合多个字段，构成一个总的字符串。

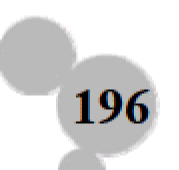
例如，把 tb_mrbook 表中的书名（bookname）和价格（price）合并到一起，构成一个新的字符串。代码如下：

```
select id,concat(bookname,":",price) as info,f_time,type from tb_mrbook;
```

其中合并后的字段名为 CONCAT 函数形成的表达式“concat(bookname,":",price)”，看上去十分复杂，通过 AS 关键字给合并字段取一个别名，这样看上去就很清晰。

9．LIMIT 子句

LIMIT 子句可以对查询结果的记录条数进行限定，控制它输出的行数。



例如，查询 tb_mrbook 表，按照图书价格降序排列，显示 3 条记录。代码如下：

```
select * from tb_mrbook order by price desc limit 3;
```

使用 LIMIT 子句还可以从查询结果的中间部分取值。首先要定义两个参数，参数 1 是开始读取的第一条记录的编号（在查询结果中，第一个结果的记录编号是 0，而不是 1）；参数 2 是要查询记录的个数。

例如，查询 tb_mrbook 表，从编号 1 开始（即从第 2 条记录），查询 4 条记录，代码如下：

```
select * from tb_mrbook where id limit 1,4;
```

10．函数和表达式

在 MySQL 中，还可以使用表达式来计算各列的值，作为输出结果。表达式还可以包含一些函数。

例如，计算 tb_mrbook 表中各类图书的总价格，代码如下：

```
select sum(price) as total,type from tb_mrbook group by type;
```

在对 MySQL 数据库进行操作时，有时需要对数据库中的记录进行统计，如求平均值、最小值、最大值等，这时可以使用 MySQL 中的统计函数。常用的统计函数如表 7.6 所示。

表 7.6　常用的统计函数

名　　称	说　　明
avg（字段名）	获取指定列的平均值
count（字段名）	如果指定了一个字段，则会统计出该字段中的非空记录；如果在前面增加 DISTINCT，则会统计不同值的记录，相同的值当作一条记录；如果使用 COUNT（*），则统计包含空值的所有记录数
min（字段名）	获取指定字段的最小值
max（字段名）	获取指定字段的最大值
std（字段名）	指定字段的标准背离值
stdtev（字段名）	与 STD 相同
sum（字段名）	指定字段所有记录的总和

除了使用函数之外，还可以使用算术运算符、字符串运算符，以及逻辑运算符来构成表达式。

例如，可以计算图书打 8 折之后的价格，代码如下：

```
select *, (price * 0.8) as '80%' from tb_mrbook;
```

7.6　MySQL 数据类型

在 MySQL 数据库中，每一条数据都有其数据类型。MySQL 支持的数据类型主要包括 3 类，即数字类型、字符串（字符）类型、日期和时间类型。

7.6.1　数字类型

MySQL 支持所有的 ANSI/ISO SQL 92 数字类型。这些类型包括准确数字的数据类型（NUMERIC、DECIMAL、INTEGER 和 SMALLINT），还包括近似数字的数据类型（FLOAT、REAL 和 DOUBLE PRECISION）。其中的关键词 INT 是 INTEGER 的同义词，关键词 DEC 是 DECIMAL 的同义词。

数字类型总体可以分成整型和浮点型两类，详细内容如表 7.7 和表 7.8 所示。

表 7.7　整型数据类型

数据类型	取值范围	说　明	单　位
TINYINT	符号值：−127～127 无符号值：0～255	最小的整数	1 字节
BIT	符号值：−127～127 无符号值：0～255	最小的整数	1 字节
BOOL	符号值：−127～127 无符号值：0～255	最小的整数	1 字节
SMALLINT	符号值：−32768～32767 无符号值：0～65535	小型整数	2 字节
MEDIUMINT	符号值：−8388608～8388607 无符号值：0～16777215	中型整数	3 字节
INT	符号值：−2147683648～2147683647 无符号值：0～4294967295	标准整数	4 字节
BIGINT	符号值： −9223372036854775808～9223372036854775807 无符号值：0～18446744073709551615	大整数	8 字节

表 7.8　浮点型数据类型

数据类型	取值范围	说　明	单　位
FLOAT	+(−)3.402823466E+38	单精度浮点数	8 或 4 字节
DOUBLE	+(−)1.7976931348623157E+308 +(−)2.2250738585072014E−308	双精度浮点数	8 字节
DECIMAL	可变	一般整数	自定义长度

技巧：在创建表时，使用哪种数字类型，应遵循以下原则。

（1）选择最小的可用类型，如果值永远不超过 127，则使用 TINYINT 比 INT 强。

（2）对于完全都是数字的，可以选择整数类型。

（3）浮点类型用于可能具有小数部分的数，如货物单价、网上购物交付金额等。

7.6.2　字符串类型

字符串类型可以分为 3 类，即普通的文本字符串类型（CHAR 和 VARCHAR）、可变类型（TEXT 和 BLOB）和特殊类型（SET 和 ENUM）。它们之间都有一定的区别，取值的范围不同，应用的地方也不同。

（1）普通的文本字符串类型，即 CHAR 和 VARCHAR 类型。CHAR 列的长度被固定为创建表所声明的长度，取值在 1～255 之间；VARCHAR 列的值是变长的字符串，取值和 CHAR 一样。普通的文本字符串类型如表 7.9 所示。

（2）可变类型（TEXT 和 BLOB）。它们的大小可以改变，TEXT 类型适合存储长文本，而 BLOB 类型适合存储二进制数据，支持任何数据，如文本、声音和图像等。TEXT 和 BLOB 类型如表 7.10 所示。



表 7.9　普通的文本字符串类型

类　　型	取 值 范 围	说　　明
[national] char(M) [binary|ASCII|unicode]	0～255 个字符	固定长度为 M 的字符串，其中 M 的取值范围为 0～255。national 关键字指定了应该使用的默认字符集。binary 关键字指定了数据是否区分大小写（默认是区分大小写的）。ASCII 关键字指定了在该列中使用 latin1 字符集。unicode 关键字指定了使用 UCS 字符集
char	0～255 个字符	与 char(M)类似
[national] varchar(M) [binary]	0～255 个字符	长度可变，其他和 char(M)类似

表 7.10　TEXT 和 BLOB 类型

类　　型	最大长度（字节数）	说　　明
TINYBLOB	2^8～1(225)	小 BLOB 字段
TINYTEXT	2^8～1(225)	小 TEXT 字段
BLOB	2^16～1(65 535)	常规 BLOB 字段
TEXT	2^16～1(65 535)	常规 TEXT 字段
MEDIUMBLOB	2^24～1(16 777 215)	中型 BLOB 字段
MEDIUMTEXT	2^24～1(16 777 215)	中型 TEXT 字段
LONGBLOB	2^32～1(4 294 967 295)	长 BLOB 字段
LONGTEXT	2^32～1(4 294 967 295)	长 TEXT 字段

（3）特殊类型（SET 和 ENUM）。

SET 和 ENUM 类型如表 7.11 所示。

表 7.11　ENUM 和 SET 类型

类　　型	最　大　值	说　　明
Enum ("value1", "value2", …)	65535	该类型的列只可以容纳所列值之一或为 null
Set ("value1", "value2", …)	64	该类型的列可以容纳一组值或为 null

技巧：在创建表时，使用字符串类型时应遵循以下原则。

（1）从速度方面考虑，要选择固定的列，可以使用 CHAR 类型。

（2）要节省空间，使用动态的列，可以使用 VARCHAR 类型。

（3）要将列中的内容限制在一种选择，可以使用 ENUM 类型。

（4）允许在一个列中有多于一个的条目，可以使用 SET 类型。

（5）如果要搜索的内容不区分大小写，可以使用 TEXT 类型。

（6）如果要搜索的内容区分大小写，可以使用 BLOB 类型。

7.6.3　日期和时间数据类型

日期和时间数据类型包括 DATETIME、DATE、TIMESTAMP、TIME 和 YEAR。其中每种类型都有其

取值的范围，如赋予它一个不合法的值，将会被“0”代替。日期和时间数据类型如表 7.12 所示。

表 7.12　日期和时间数据类型

类　型	取 值 范 围	说　明
DATE	1000-01-01　9999-12-31	日期，格式 YYYY-MM-DD
TIME	-838:58:59　835:59:59	时间，格式 HH:MM:SS
DATETIME	1000-01-01 00:00:00 9999-12-31 23:59:59	日期和时间，格式 YYYY-MM-DD HH:MM:SS
TIMESTAMP	1970-01-01 00:00:00 2037 年的某个时间	时间标签，在处理报告时使用显示格式取决于 M 的值
YEAR	1901-2155	年份可指定两位数字和 4 位数字的格式

在 MySQL 中，日期的顺序是按照标准的 ANSISQL 格式进行输出的。

7.7　phpMyAdmin 管理 MySQL 数据库

phpMyAdmin 是众多 MySQL 图形化管理工具中应用最广泛的一种，是一款使用 PHP 开发的 B/S 模式的 MySQL 客户端软件，该工具是基于 Web 跨平台的管理程序，并且支持简体中文。用户可以在官方网站“www.phpMyAdmin.net”上免费下载到最新的版本。phpMyAdmin 为 Web 开发人员提供了类似于 Access、SQL Server 的图形化数据库操作界面，通过该管理工具可以完全对 MySQL 进行操作，如创建数据库、数据表、生成 MySQL 数据库脚本文件等。

7.7.1　管理数据库

在浏览器地址栏中输入“http://localhost/phpmyadmin/”，进入 phpMyAdmin 图形化管理主界面，接下来就可以进行 MySQL 数据库的操作，下面将分别介绍如何创建、修改和删除数据库。

1．创建数据库

进入 phpMyAdmin 的主界面，首先在文本框中输入数据库的名称“db_study”，然后在下拉列表框中选择所要使用的编码，一般选择 gb2312_chinese_ci 简体中文编码格式，单击“创建”按钮，创建数据库，如图 7.18 所示。成功创建数据库后，将显示如图 7.19 所示的界面。

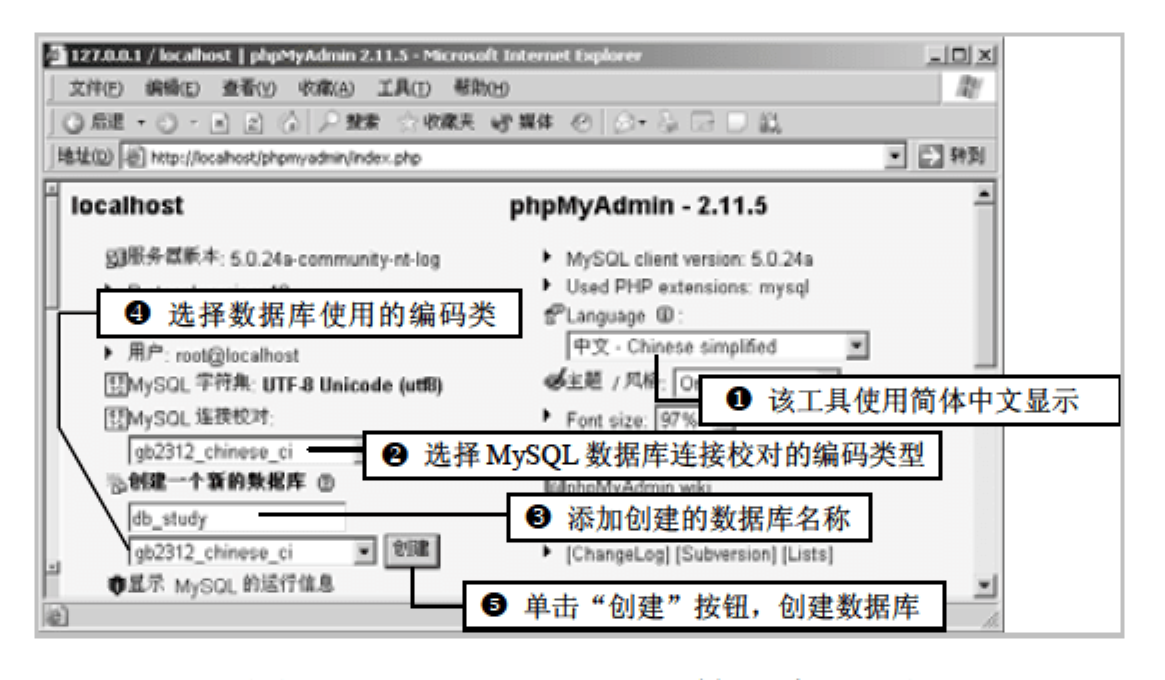


图 7.18　phpMyAdmin 管理主界面

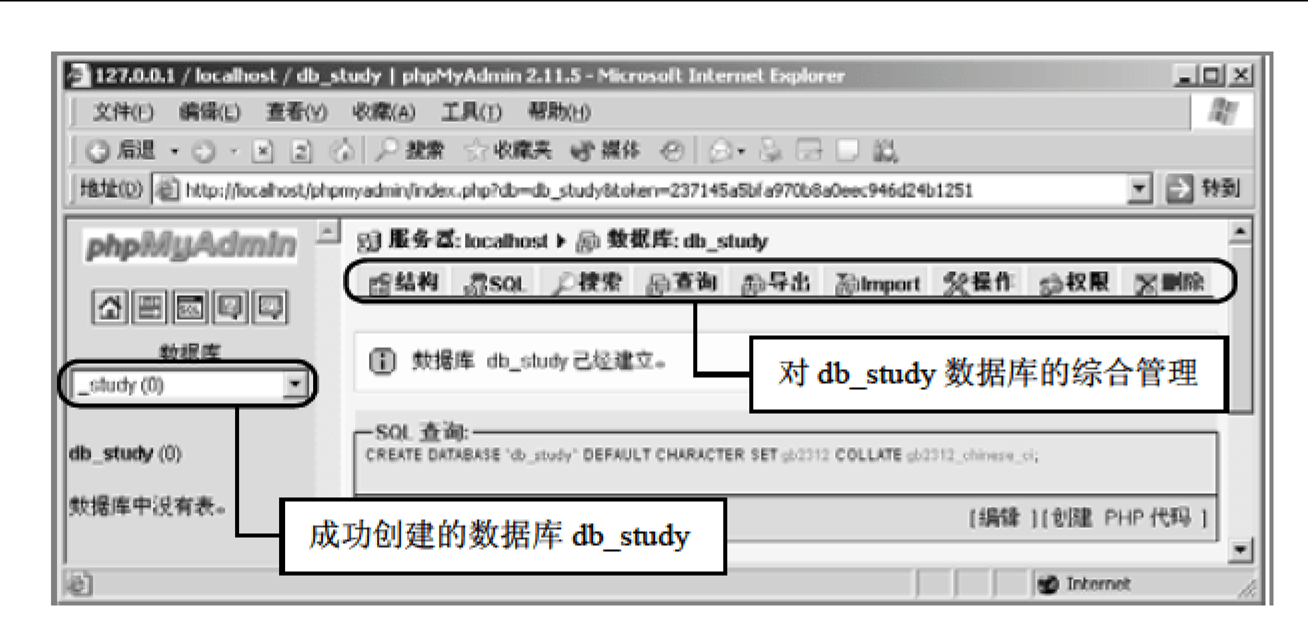


图 7.19　成功创建数据库

说明：在右侧界面中可以对数据库进行相关操作，如结构、SQL、导出、搜索、查询、删除等，单击相应的超链接即可进入相应的操作界面。但是在创建数据库后还没有创建数据表的情况下，只能够执行结构、SQL、Import、操作、权限和删除 6 项操作，其他 3 项操作（搜索、查询和导出）不能执行，当指向其超链接时，弹出不可用标记。

2．修改数据库

在如图 7.19 所示的界面中，在右侧界面还可以对当前数据库进行修改。单击界面中的操作超链接，进入修改操作页面。

（1）可以对当前数据库执行创建数据表的操作，只要在创建数据表的提示信息下面的两个文本框中分别输入要创建的数据表的名称和字段总数，然后单击“执行”按钮，即可进入到创建数据表结构页面。

（2）也可以对当前的数据库重命名，在“重新命名数据库为”下的文本框中输入新的数据库名称，单击“执行”按钮，即可成功修改数据库名称。修改数据库名称如图 7.20 所示。

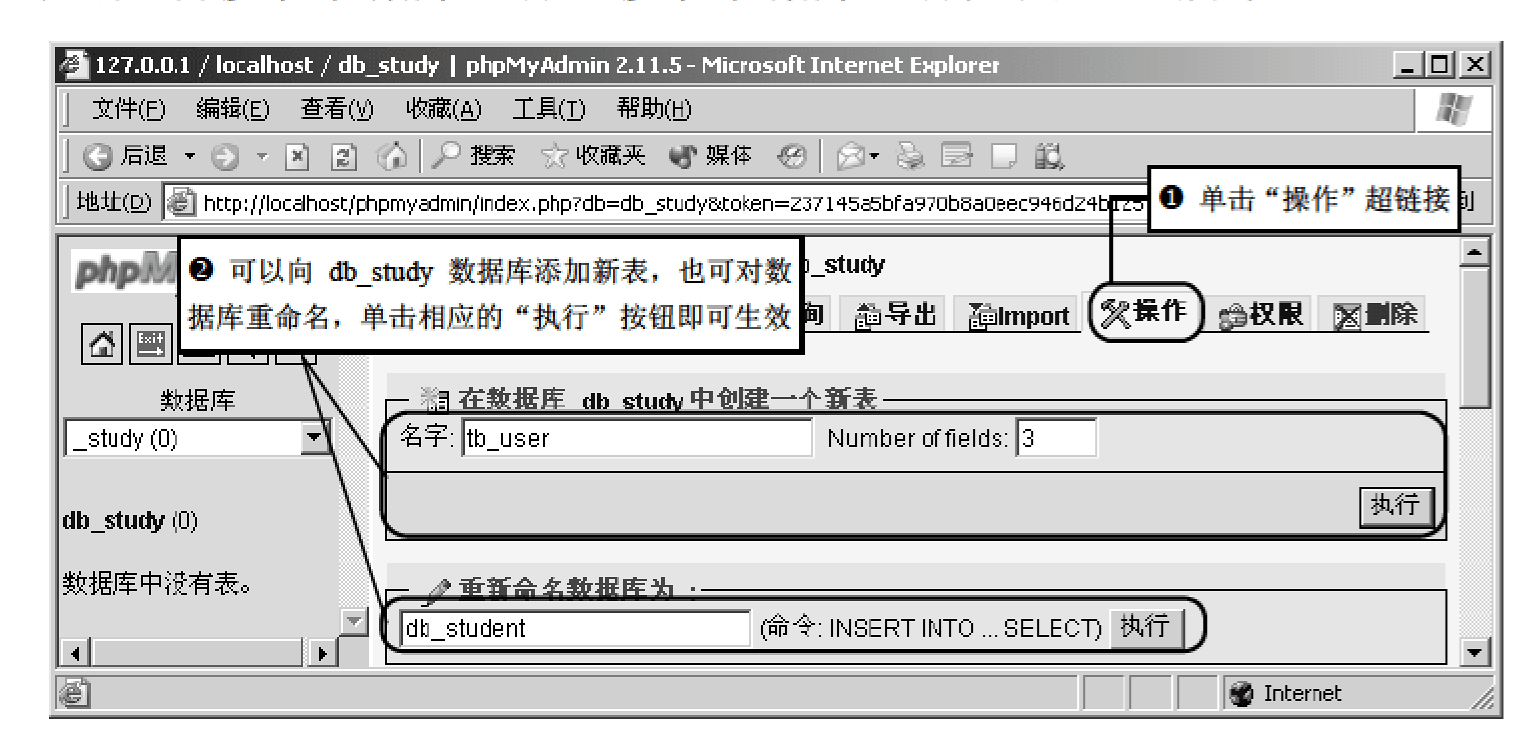


图 7.20　修改数据库

3．删除数据库

要删除某个数据库，首先在界面左侧的下拉菜单中选择该数据库，然后单击右侧界面中的删除超链接（如图 7.20 所示），即可成功删除指定的数据库。

7.7.2　管理数据表

管理数据表是以选择指定的数据库为前题，然后在该数据库中创建并管理数据表。下面就来介绍如何创建、修改、删除数据表。

1．创建数据表

创建数据库 db_study 后，在右侧的操作页面中输入数据表的名称和字段数，然后单击“执行”按钮，即可创建数据表，如图 7.21 所示。

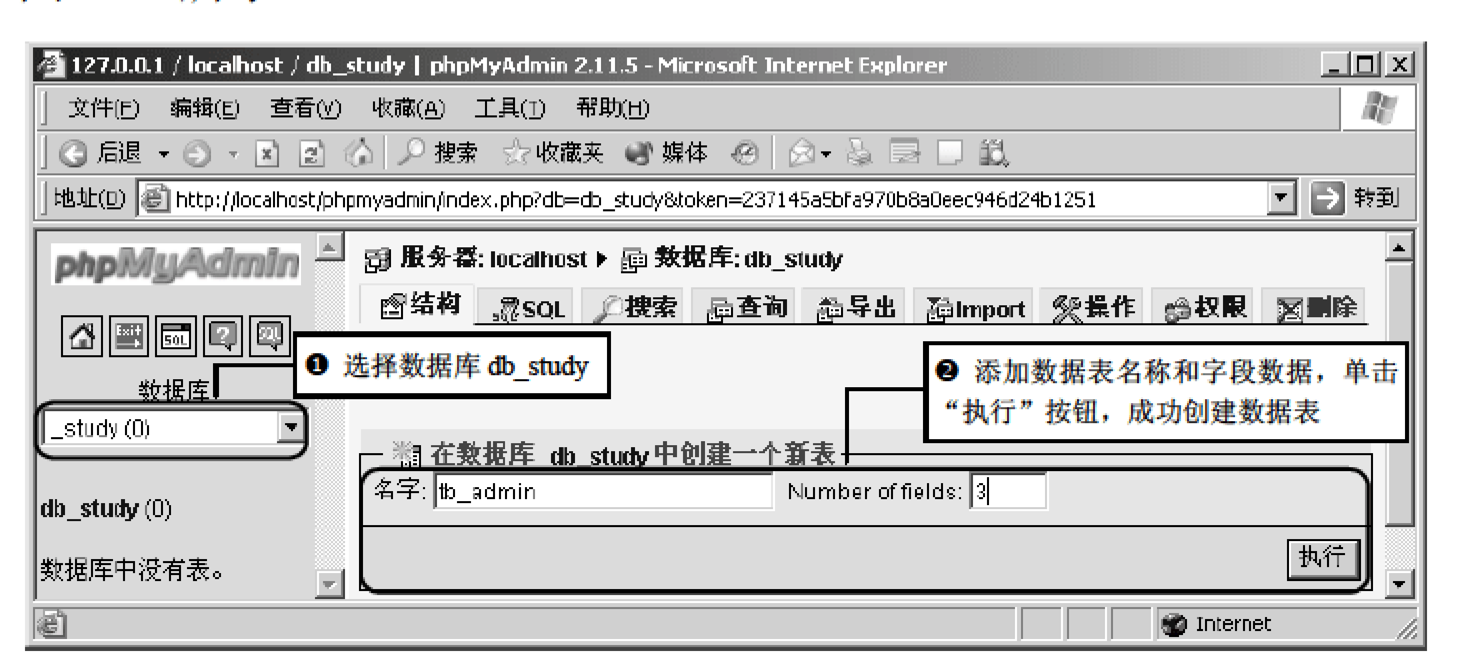


图 7.21　创建数据表

成功创建数据表 tb_admin 后，将显示数据表结构界面。在表单中对各个字段的详细信息进行录入，包括字段名、数据类型、长度/值、编码格式、是否为空、主键等，以完成对表结构的详细设置。当所有的信息都输入以后，单击“保存”按钮，创建数据表结构，如图 7.22 所示。成功创建数据表结构后，将显示如图 7.23 所示的界面。

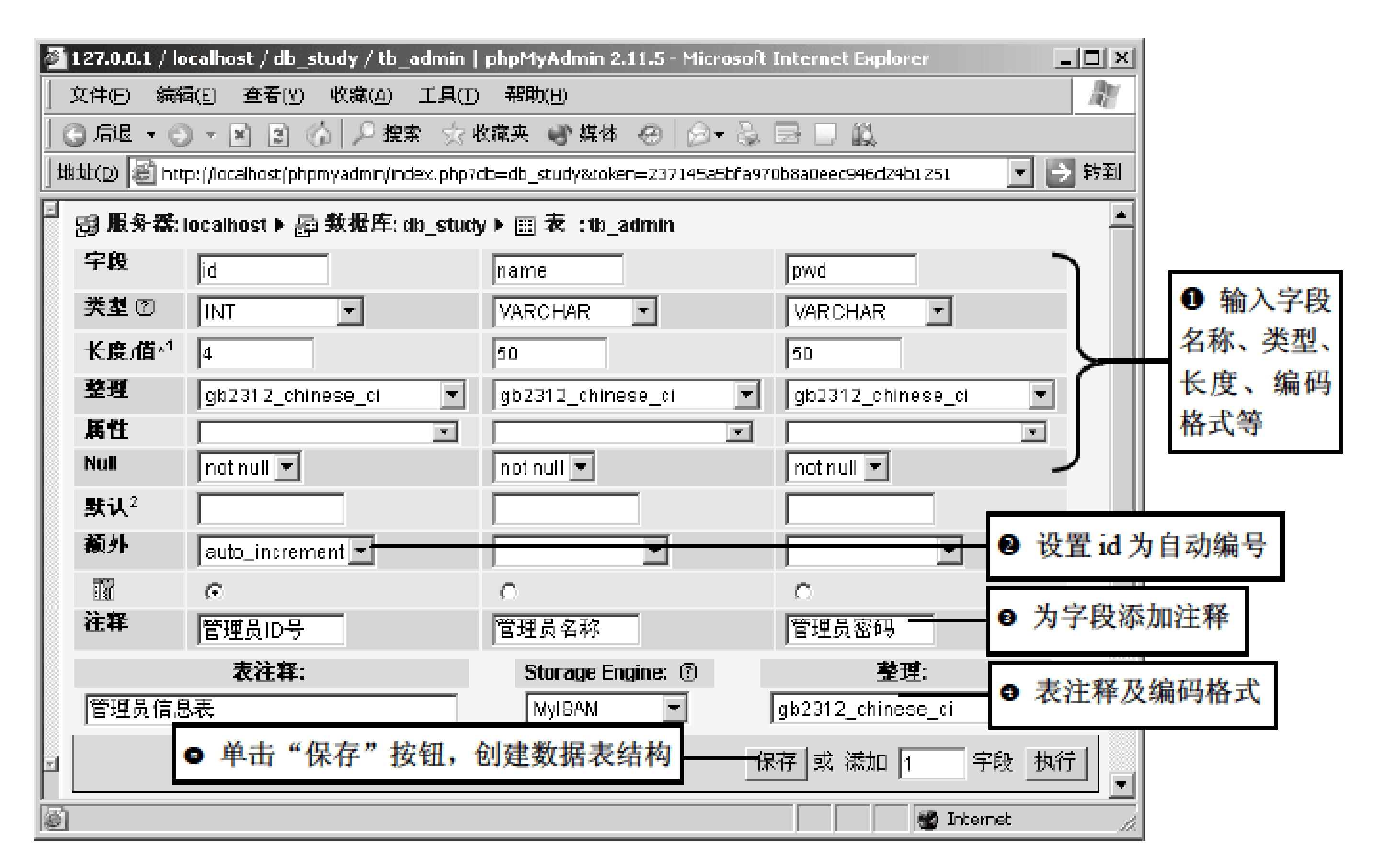


图 7.22　创建数据表结构

技巧： 单击“执行”按钮，可以对数据表结构以横版显示进行表结构编辑。

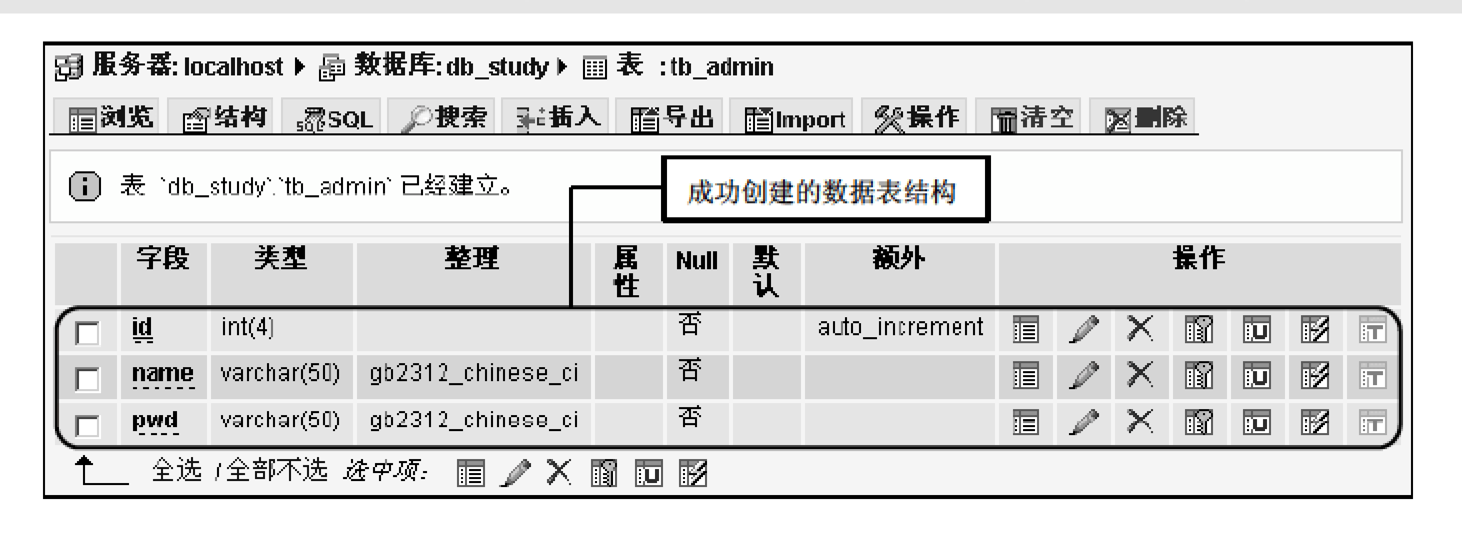


图 7.23　成功创建数据表



2．修改数据表

一个新的数据表被创建后，进入到数据表页面中，在这里可以通过改变表的结构来修改表，可以执行添加新的列、删除列、索引列、修改列的数据类型或者字段的长度/值等操作，如图 7.24 所示。



图 7.24　修改数据表结构

3．删除数据表

要删除某个数据表，首先在左侧的下拉菜单中选择该数据库，在指定的数据库中选择要删除的数据表，然后单击右侧界面中的删除超链接（如图 7.24 所示），即可成功删除指定的数据表。

7.7.3　管理数据记录

单击 phpMyAdmin 主界面中的SQL超链接，打开 SQL 语句编辑区。在编辑区输入完整的 SQL 语句，来实现数据的插入、修改、查询和删除操作。

1．使用 SQL 语句插入数据

在 SQL 语句编辑区应用 insert 语句向数据表 tb_admin 中插入数据后，单击“执行”按钮，向数据表中插入一条数据，如图 7.25 所示。如果提交的 SQL 语句有错误，系统会给出一个警告，提示用户修改它；如果提交的 SQL 语句正确，则弹出如图 7.26 所示的提示信息。

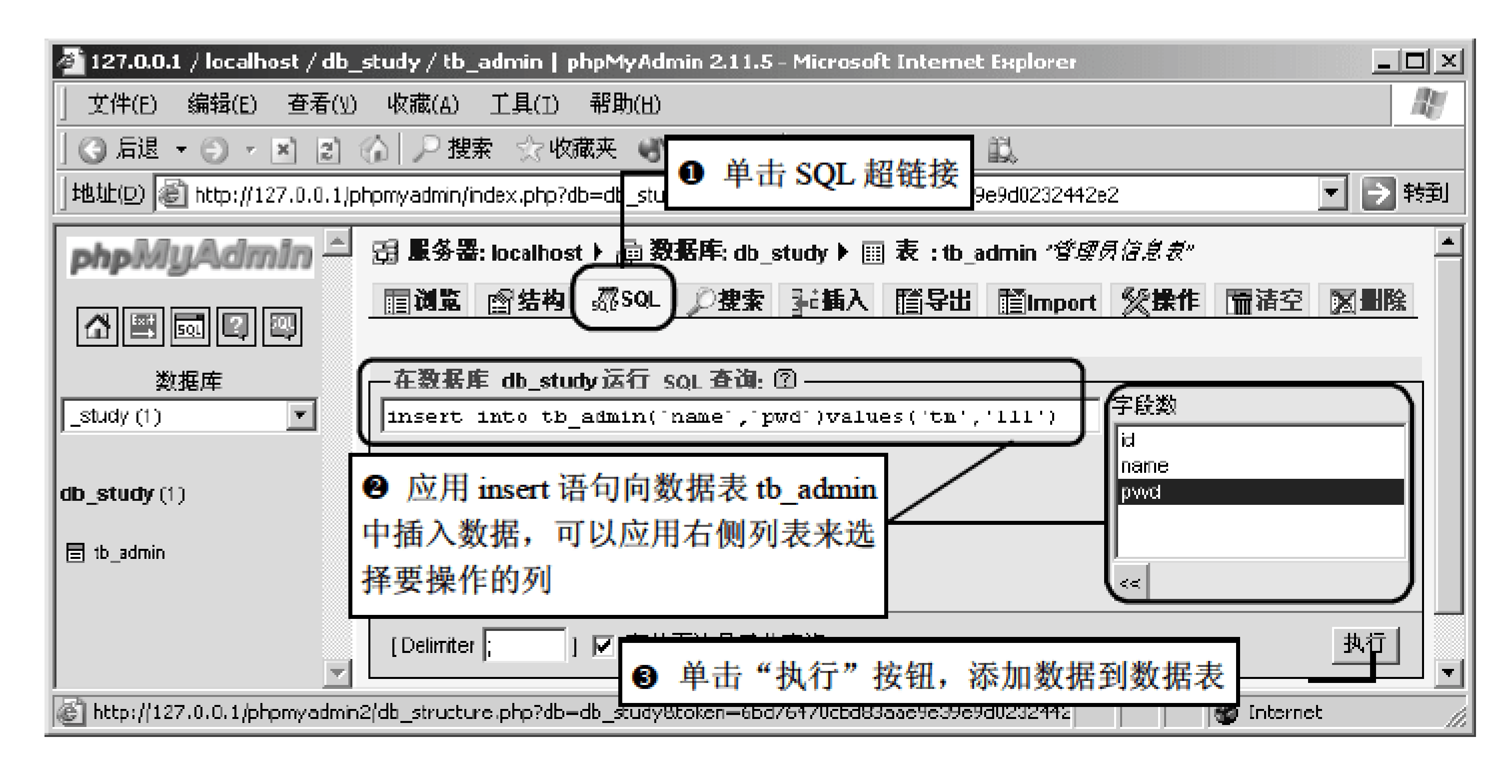


图 7.25　使用 SQL 语句向数据表中插入数据



图 7.26　成功添加数据信息

✍ **技巧**：为了编写方便，可以利用其右侧的属性列表来选择要操作的列，只要选中要添加的列，双击其选项或者单击<<按钮可添加列名称。

2．使用 SQL 语句修改数据

在 SQL 语句编辑区应用 update 语句修改数据信息，将 id 为 1 的管理员的名称改为“纯净水”，密码改为“111”，添加的 SQL 语句如图 7.27 所示。

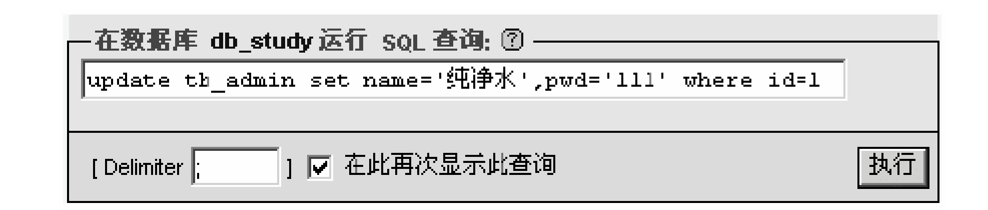


图 7.27　添加修改数据信息的 SQL 语句

单击“执行”按钮，数据修改成功。比较修改前后的数据，如图 7.28 所示。

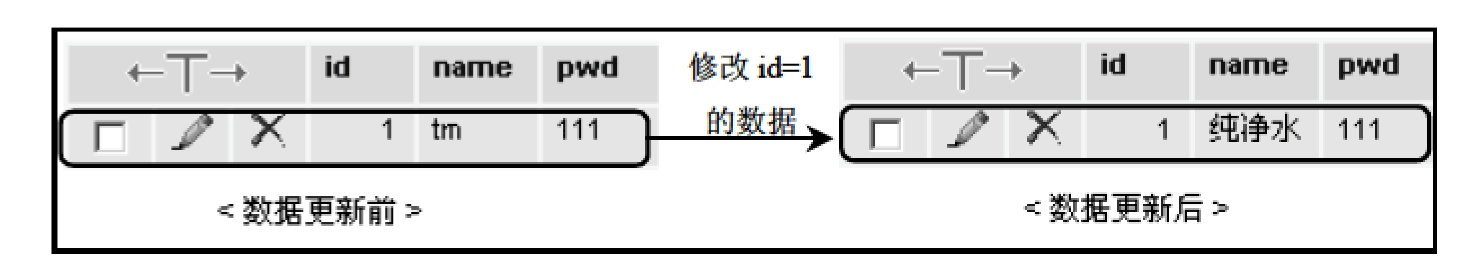


图 7.28　修改单条数据的实现过程

3．使用 SQL 语句查询数据

在 SQL 语句编辑区应用 select 语句检索指定条件的数据信息，将 id 小于 4 的管理员全部显示出来，添加的 SQL 语句如图 7.29 所示。

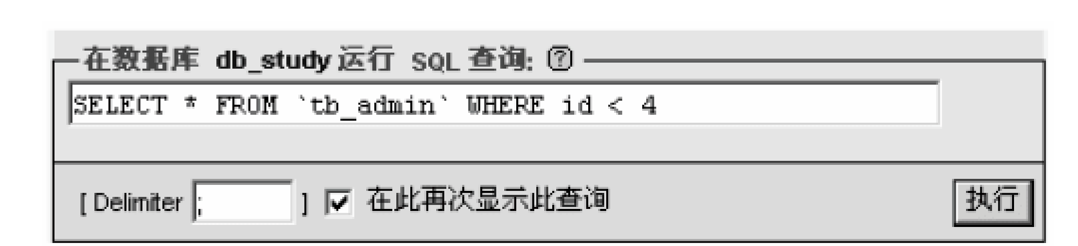


图 7.29　添加查询数据信息的 SQL 语句

单击“执行”按钮，该语句的实现过程如图 7.30 所示。

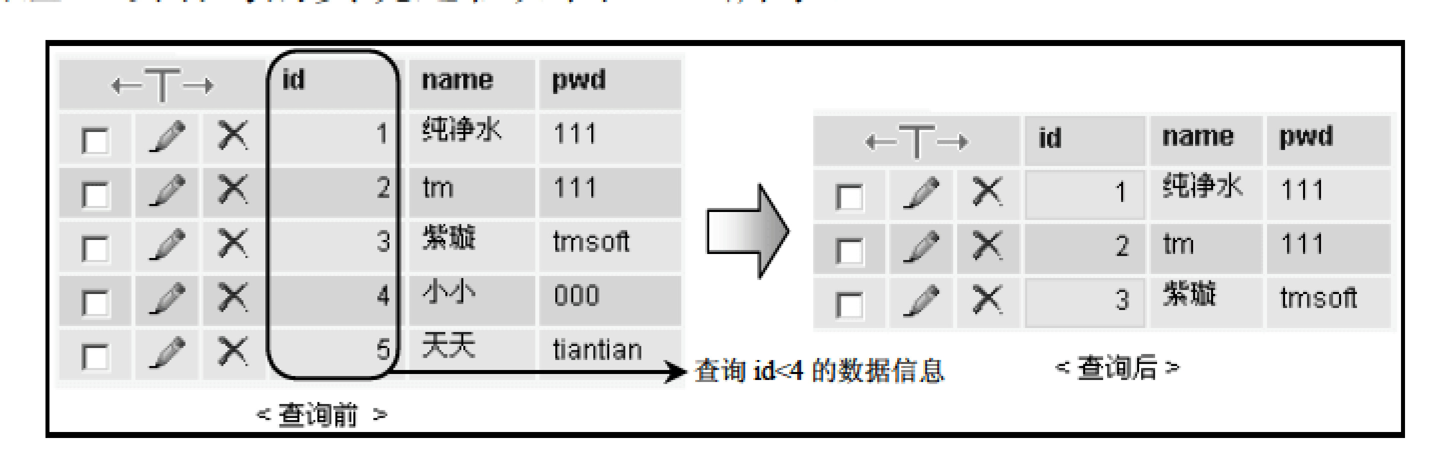


图 7.30　查询指定条件的数据信息的实现过程

除了对整个表的简单查询外，还可以执行复杂的条件查询（使用 where 子句提交 LIKE、ORDER BY、GROUP BY 等条件查询语句）及多表查询，读者可通过上机进行实践，灵活运用 SQL 语句功能。

4．使用 SQL 语句删除数据

在 SQL 语句编辑区应用 delete 语句检索指定条件的数据或全部数据信息，删除名称为 tm 的管理员信息，添加的 SQL 语句如图 7.31 所示。



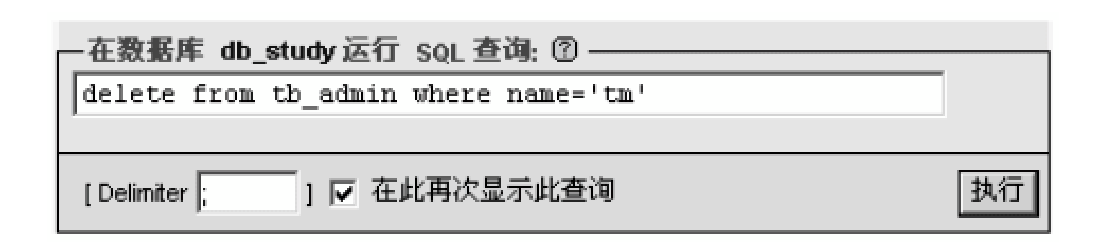


图 7.31　添加删除指定数据信息的 SQL 语句

注意：如果 delete 语句后面没有 where 条件值，那么将删除指定数据表中的全部数据。

单击“执行”按钮，弹出确认删除操作对话框，单击“确定”按钮，执行数据表中指定条件的删除操作。该语句的实现过程如图 7.32 所示。

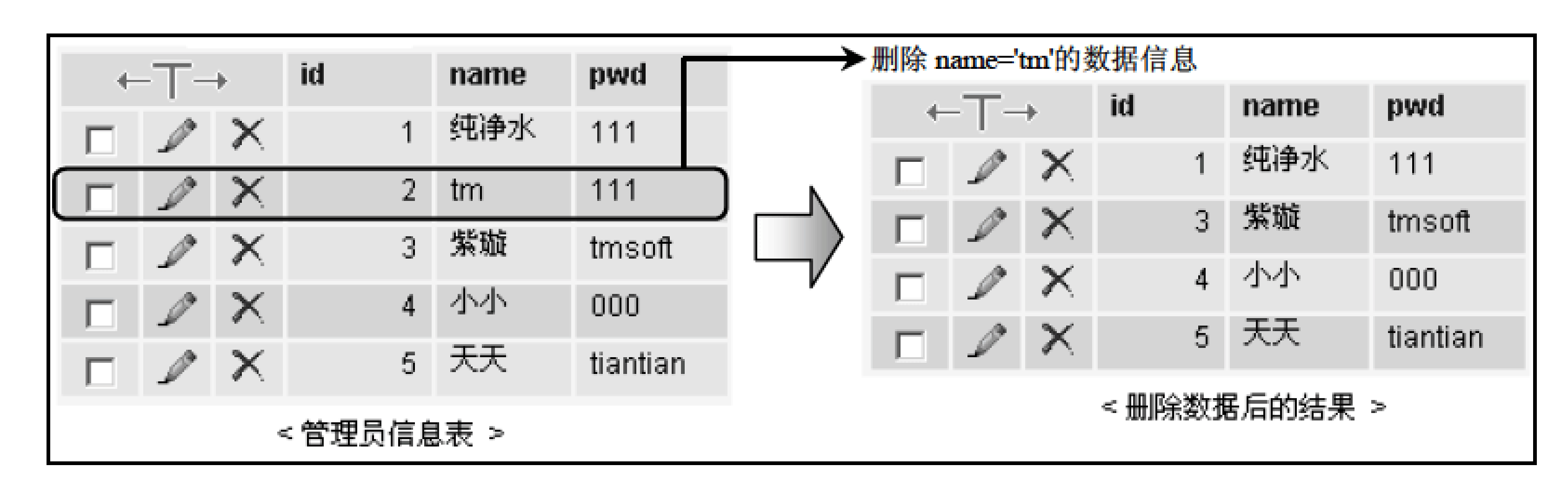


图 7.32　删除指定条件的数据信息的实现过程

5．通过 form 表单插入数据

选择某个数据表后，单击插入超链接，进入插入数据界面，如图 7.33 所示。在界面中输入各字段值，单击“执行”按钮即可插入记录。默认情况下，一次可以插入两条记录。

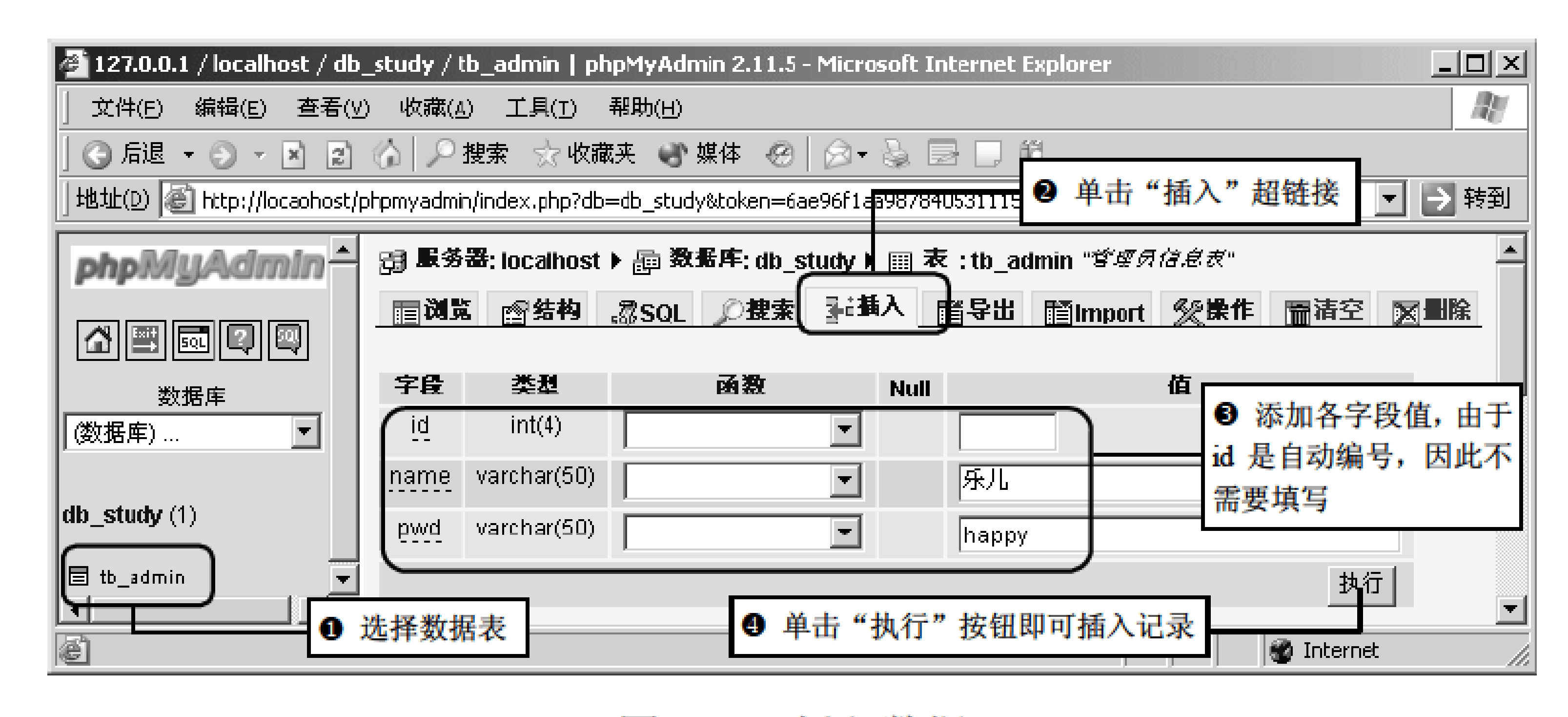


图 7.33　插入数据

6．浏览数据

选择某个数据表后，单击浏览超链接，进入浏览界面，如图 7.34 所示。单击每行记录中的✎按钮，可以对该记录进行编辑；单击每行记录中的✕按钮，可以删除该条记录。

7．搜索数据

选择某个数据表后，单击搜索超链接，进入搜索页面，如图 7.35 所示。在这个页面中，可以在选择字段的列表框中选择一个或多个列，如果要选择多个列，先按下 Ctrl 键并单击要选择的字段名，查询结果将按照选择的字段名进行输出。

在该界面中可以对记录按条件进行查询。查询方式有两种：第一种方式为选择构建 where 语句查询，直

接在“where 语句的主体”文本框中输入查询语句，然后单击其后的“执行”按钮；第二种方式使用按例查询，选择查询的条件，并在文本框中输入要查询的值，单击“执行”按钮。

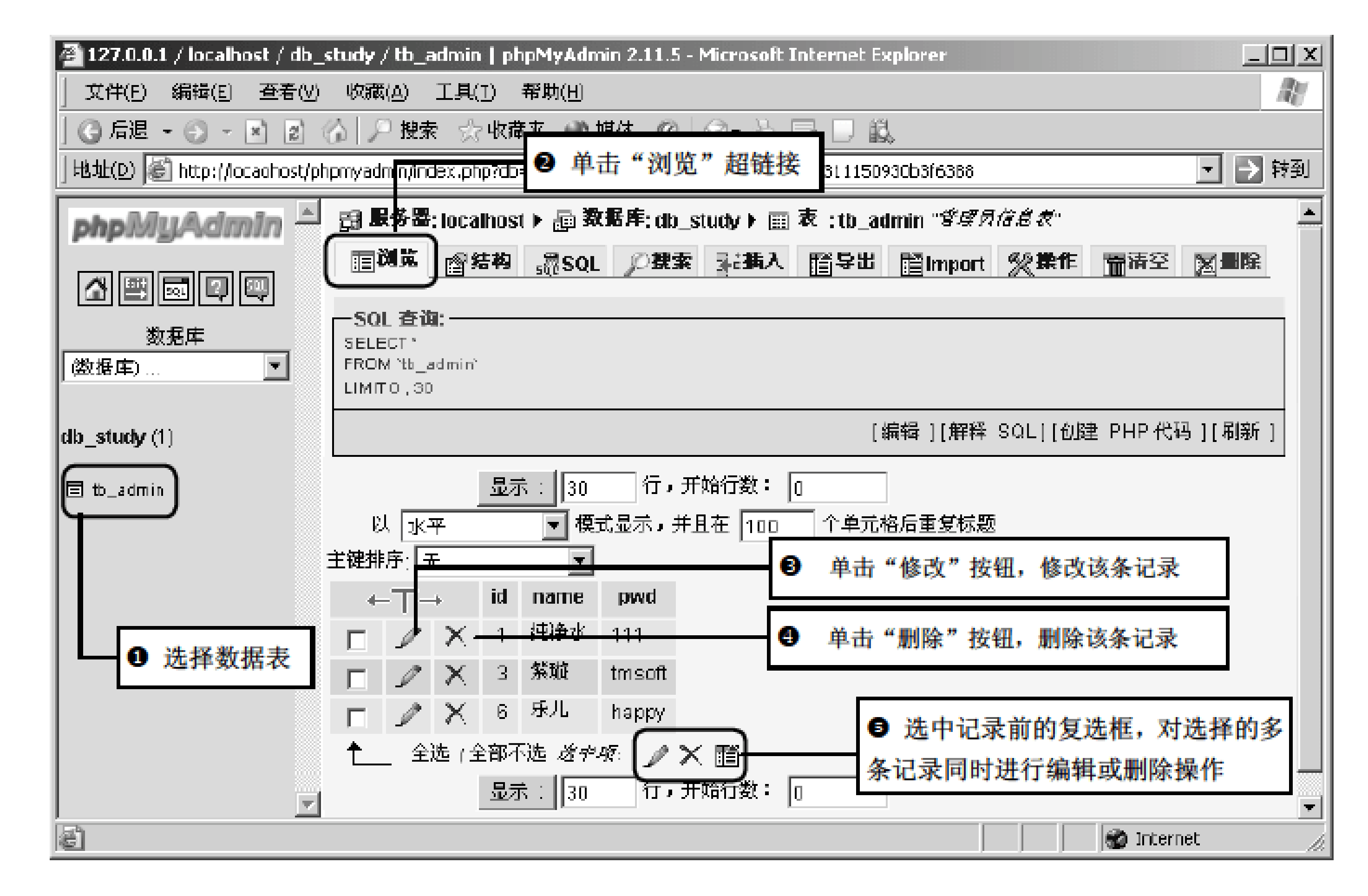


图 7.34 浏览数据

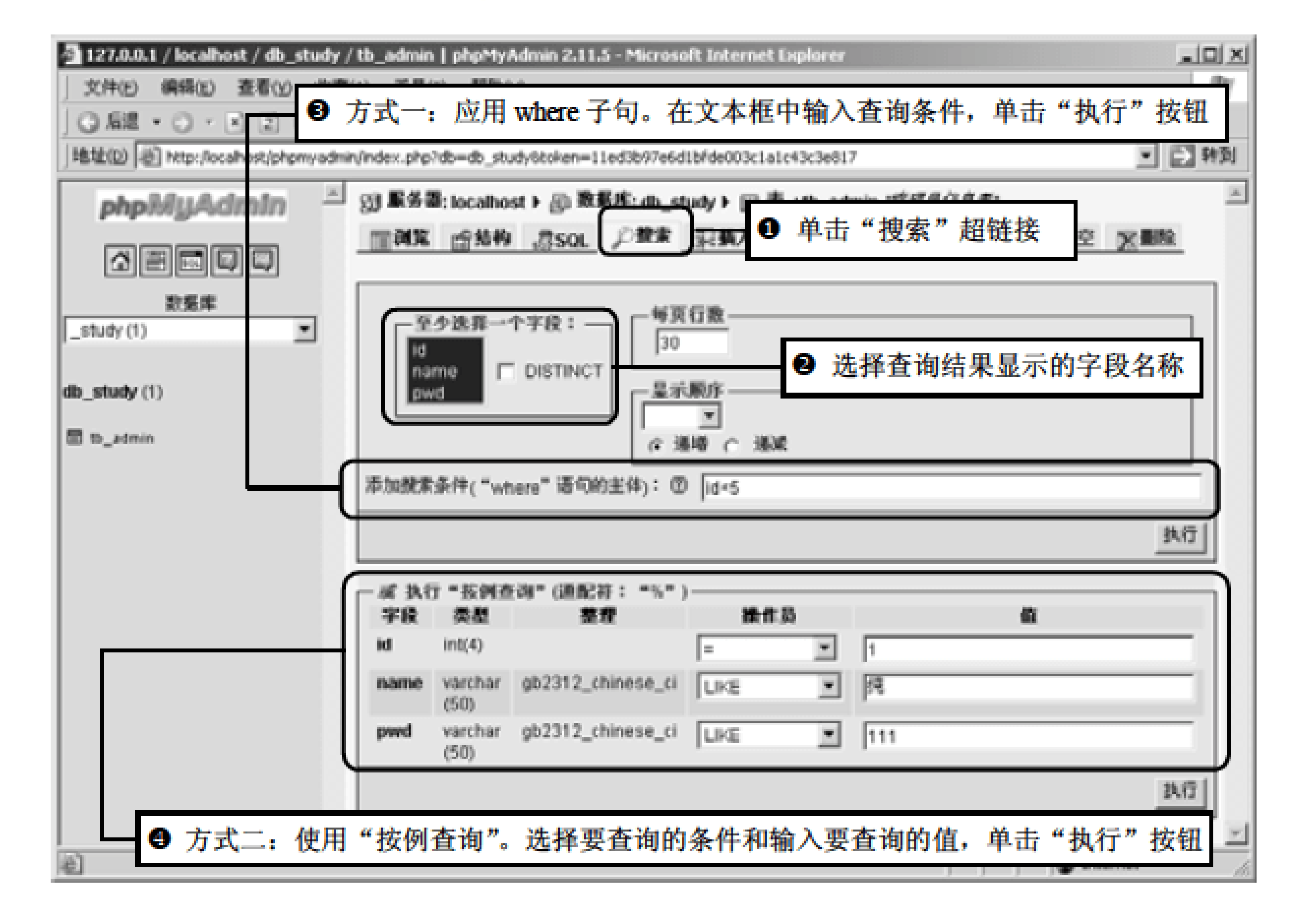


图 7.35 搜索查询

7.7.4 导入导出数据

导入和导出 MySQL 数据库脚本是互逆的两个操作。导入是执行扩展名为.sql 的文件，将数据导入到数据库中；导出是将数据表结构、表记录存储为.sql 的脚本文件。通过导入和导出的操作实现数据库的备份和还原。

1. 导出 MySQL 数据库脚本

单击 phpMyAdmin 主界面中的 导出 超链接，打开导出编辑区，如图 7.36 所示。选择导出文件的格式，这里默认使用选项 SQL，选中“另存为文件”复选框，单击“执行”按钮，弹出如图 7.37 所示的“文件下载”对话框，单击“保存”按钮，将脚本文件以.sql 格式存储在指定位置。



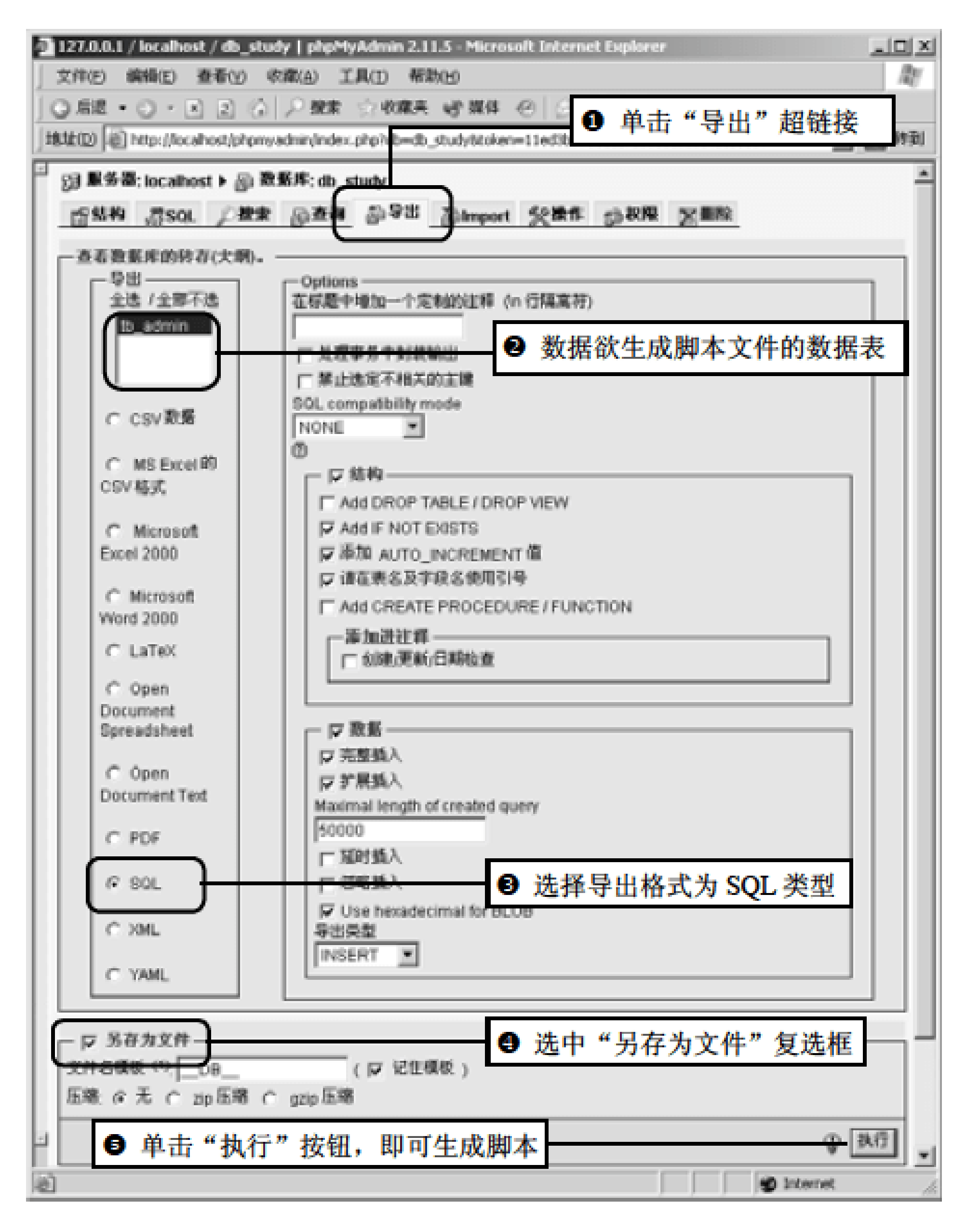


图 7.36　生成 MySQL 脚本文件设置界面

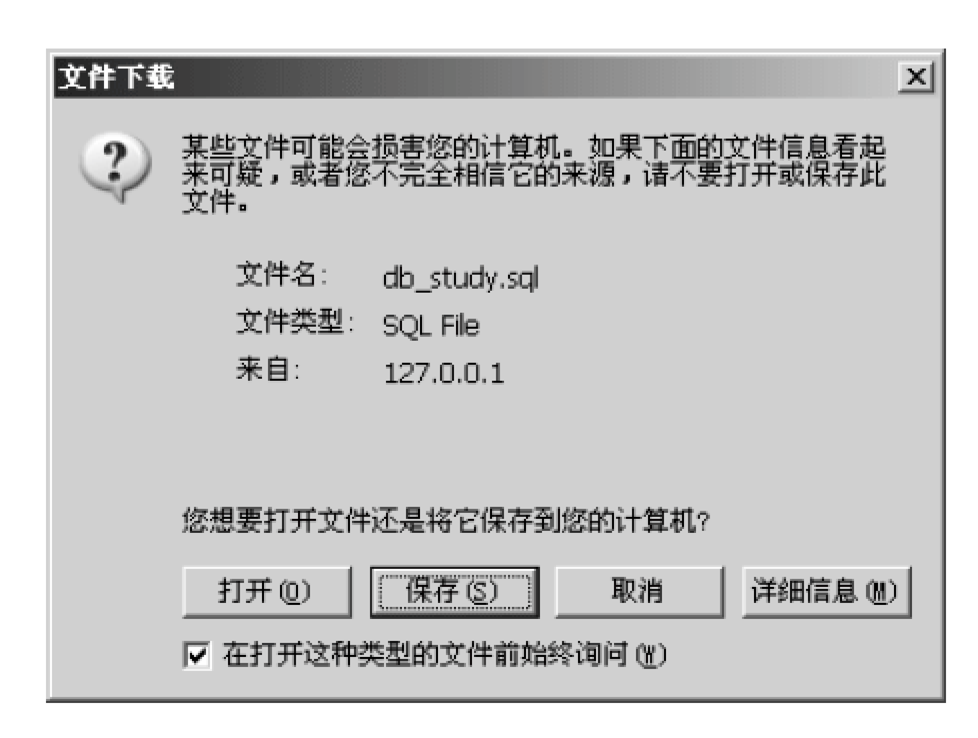


图 7.37　“文件下载”对话框

2．导入 MySQL 数据库脚本

单击 Import 超链接，进入执行 MySQL 数据库脚本界面，单击“浏览”按钮查找脚本文件（如 db_study.sql）所在位置，如图 7.38 所示，单击“执行”按钮，即可执行 MySQL 数据库脚本文件。

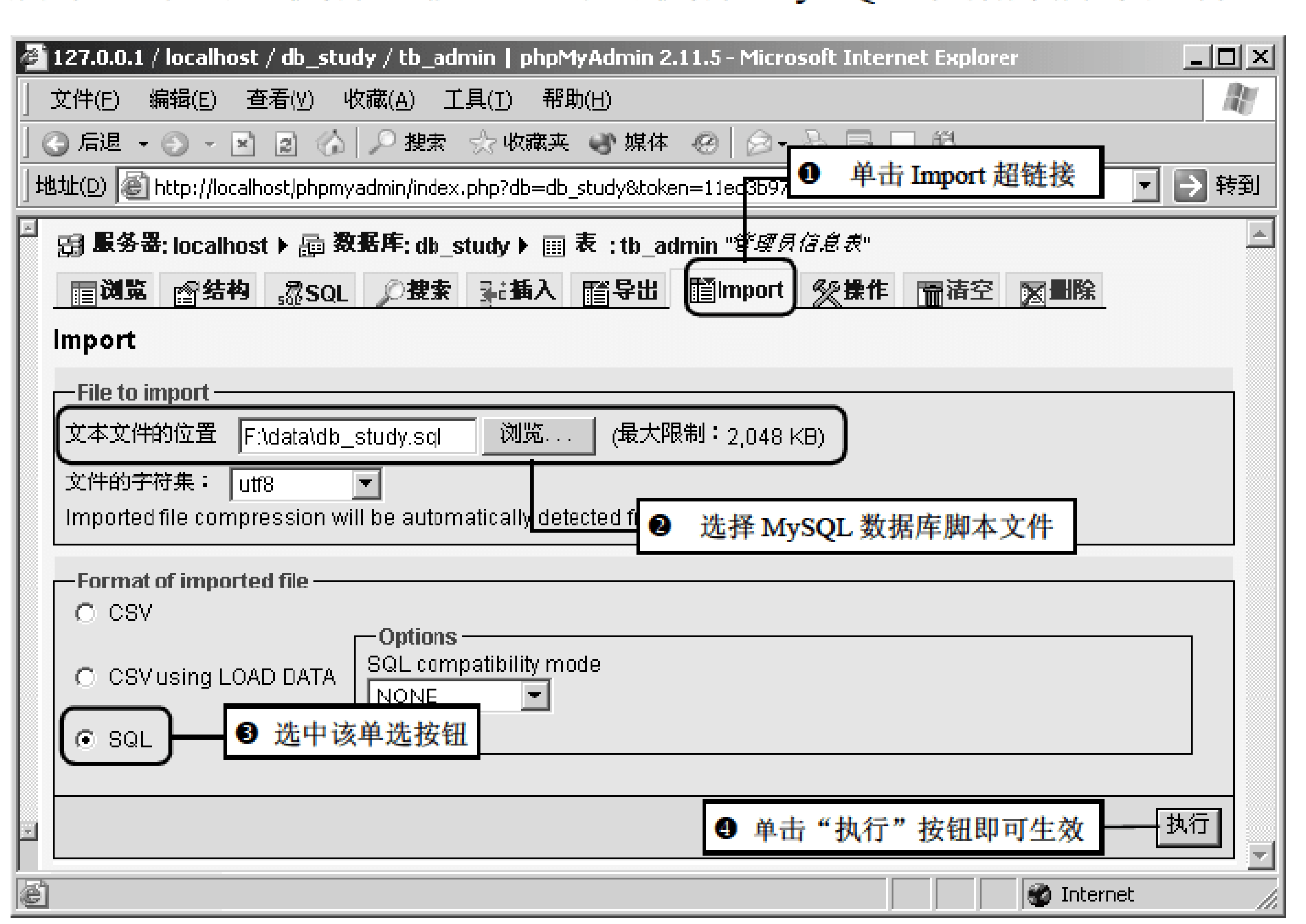


图 7.38　执行 MySQL 数据库脚本文件

注意：在执行 MySQL 脚本文件前，首先检测是否有与所导入数据库同名的数据库，如果没有，则首先要在数据库中创建一个名称与数据文件中的数据库名相同的数据库，然后再执行 MySQL 数据库脚本文件。另外，在当前数据库中，不能有与将要导入数据库中的数据表重名的数据表存在，如果有重名的表存在，导入文件就会失败，提示错误信息。

说明：读者也可通过单击 phpMyAdmin 图形化工具左侧区的按钮，在打开的对话框中单击“导入文件”超链接，然后选择脚本文件所在的位置，从而执行脚本文件。

7.8　照猫画虎——基本功训练

7.8.1　基本功训练 1——创建 db_book 图书数据库

视频讲解：光盘\mr\07\lx\创建 db_book 图书数据库.exe

登录 MySQL 数据库后，可以通过 create database 命令创建新的 MySQL 数据库。这里使用 create database 命令创建名为 db_book 的图书数据库。其实现过程如下：

（1）进入命令提示符窗口，启动 MySQL 服务，连接 MySQL 服务器。

（2）通过数据库函数在命令提示符下编写命令，代码如下：

```
create database db_book;
```

按 Enter 键，运行结果如图 7.39 所示。

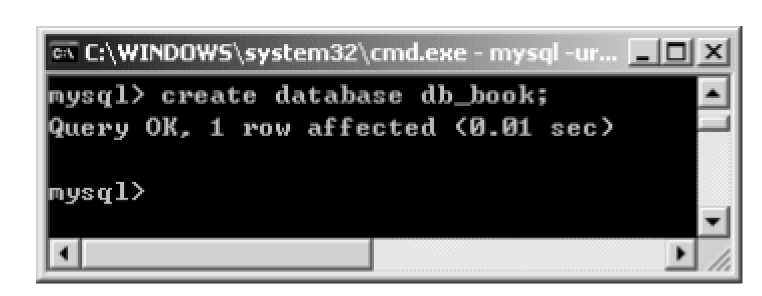


图 7.39　创建 MySQL 数据库

照猫画虎：查看上面创建的 db_book 数据库。提示：使用 show databases db_book 命令。（20 分）

7.8.2　基本功训练 2——在 db_book 数据库中创建图书信息表 tb_mrbook

视频讲解：光盘\mr\07\lx\在 db_book 数据库中创建图书信息表 tb_mrbook.exe

数据库的主要作用是用来保存数据表，而数据表的作用是保存数据信息。本实例通过 CREATE TABLE 语句在图书数据库中创建图书信息表，操作命令如图 7.40 所示。

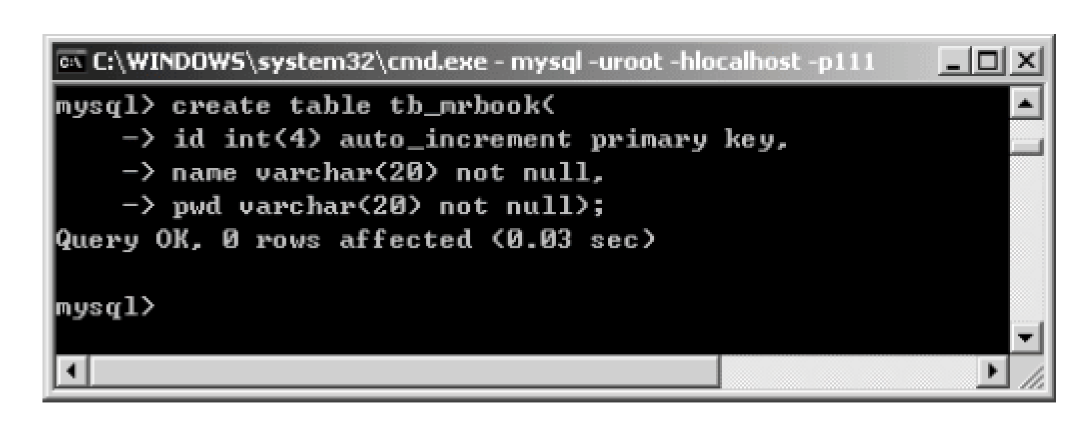


图 7.40　数据表创建成功

实现过程如下：

（1）启动 MySQL 服务，建立 MySQL 服务器连接。

（2）使用“use db_book;”语句设定数据库标记，这里设定的数据库是 db_book 数据库。

（3）通过“create table tb_mrbook();”语句创建数据表，代码如下：

```
MySQL> use db_book;
Database changed
```



```
MySQL> create table tb_mrbook(
    -> id int(4) auto_increment primary key,
    -> name varchar(20) NOT NULL,
    -> pwd varchar(20) NOT NULL);
Query OK, 0 rows affected (0.05 sec)
```

（4）断开 MySQL 服务器连接，结束 MySQL 服务。

注意：在创建数据表的过程中，一定要注意尖括号（<）和括号的区别。把二者弄混，会导致数据表创建失败。

照猫画虎：查看 tb_mrbook 图书信息表结构。提示：可以使用 show columns from 语句查看表的结构。（20 分）

7.8.3　基本功训练 3——向 tb_mrbook 表中添加数据

视频讲解：光盘\mr\07\lx\向 tb_mrbook 表中添加数据.exe

建立数据库是为了把不同类型或者不同要求的数据分类存储在不同的数据表中，创建数据表是为了将相同类型或者相同要求的数据存储在同一表中。本实例讲解通过 SQL 语句向图书信息表中添加数据，运行结果如图 7.41 所示。

实现过程如下：

（1）在命令提示符窗口下，开启 MySQL 服务，通过命令行语句连接 MySQL 服务器，选择数据库和数据表。

（2）编辑 insert 语句，执行数据的添加操作，代码如下：

```
insert into tb_mrbook_php(id,name,pwd,last_date)values ('','lizhonghua','5315',now());
```

（3）当提示符窗口出现如图 7.42 所示的内容，表示数据添加成功。

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -hlocalhost -p111
mysql> insert into tb_mrbook
    -> (id,name,pwd)values
    -> ('','yangming','5315');
Query OK, 1 row affected, 1 warning (0.02 sec)

mysql> insert into tb_mrbook
    -> (id,name,pwd)values
    -> ('','pankaihua','8023');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into tb_mrbook
    -> (id,name,pwd)values
    -> ('','lihui','571');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> select * from tb_mrbook;
+----+-----------+------+
| id | name      | pwd  |
+----+-----------+------+
|  1 | yangming  | 5315 |
|  2 | pankaihua | 8023 |
|  3 | lihui     | 571  |
+----+-----------+------+
3 rows in set (0.00 sec)

mysql>
```

图 7.41　向图书信息表中添加数据

```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -hlocalhost -p111
ysql> insert into tb_mrbook
    -> (id,name,pwd)values
    -> ('','lihui','571');
uery OK, 1 row affected, 1 warning (0.00 sec)
```

图 7.42　数据添加成功

（4）通过 select 语句查询数据表中内容，代码如下，显示结果如图 7.41 所示。

```
select * from tb_mrbook;
```

（5）断开 MySQL 服务器连接，关闭 MySQL 服务。

照猫画虎：查询 tb_mrbook 表中的所有信息，按照 id 进行降序排列，并且只显示两条记录。提示：SQL 语句为 select * from tb_mrbook order by id desc limit2。（20 分）

7.8.4　基本功训练 4——修改 tb_mrbook 表中的数据

视频讲解：光盘\mr\07\lx\修改 tb_mrbook 表中的数据.exe

用户向数据表中插入数据信息时，很有可能在手动输入的过程中将信息输入错误，这时就需要使用 SQL 语句对错误的数据进行修改。本实例通过修改语句，将图书信息表中操作数据表的时间进行修改，运行结果如图 7.43 所示。

实现过程如下：

（1）在命令提示符窗口下，开启 MySQL 服务，通过命令行语句连接 MySQL 服务器，创建数据库并建立数据表。

（2）使用 select 查询语句查询图书信息表，如图 7.43 中❶原数据表所示。

（3）使用 update 更新语句，更新数据表中数据，代码如下：

```
update tb_mrbook set pwd='123' where name='lihui';
```

（4）当提示符窗口出现如图 7.44 所示的内容，表示数据更新成功。

（5）通过 select 语句查询数据表中内容，显示结果如图 7.43 中❷修改后数据所示。

（6）断开 MySQL 服务器连接，关闭 MySQL 服务。

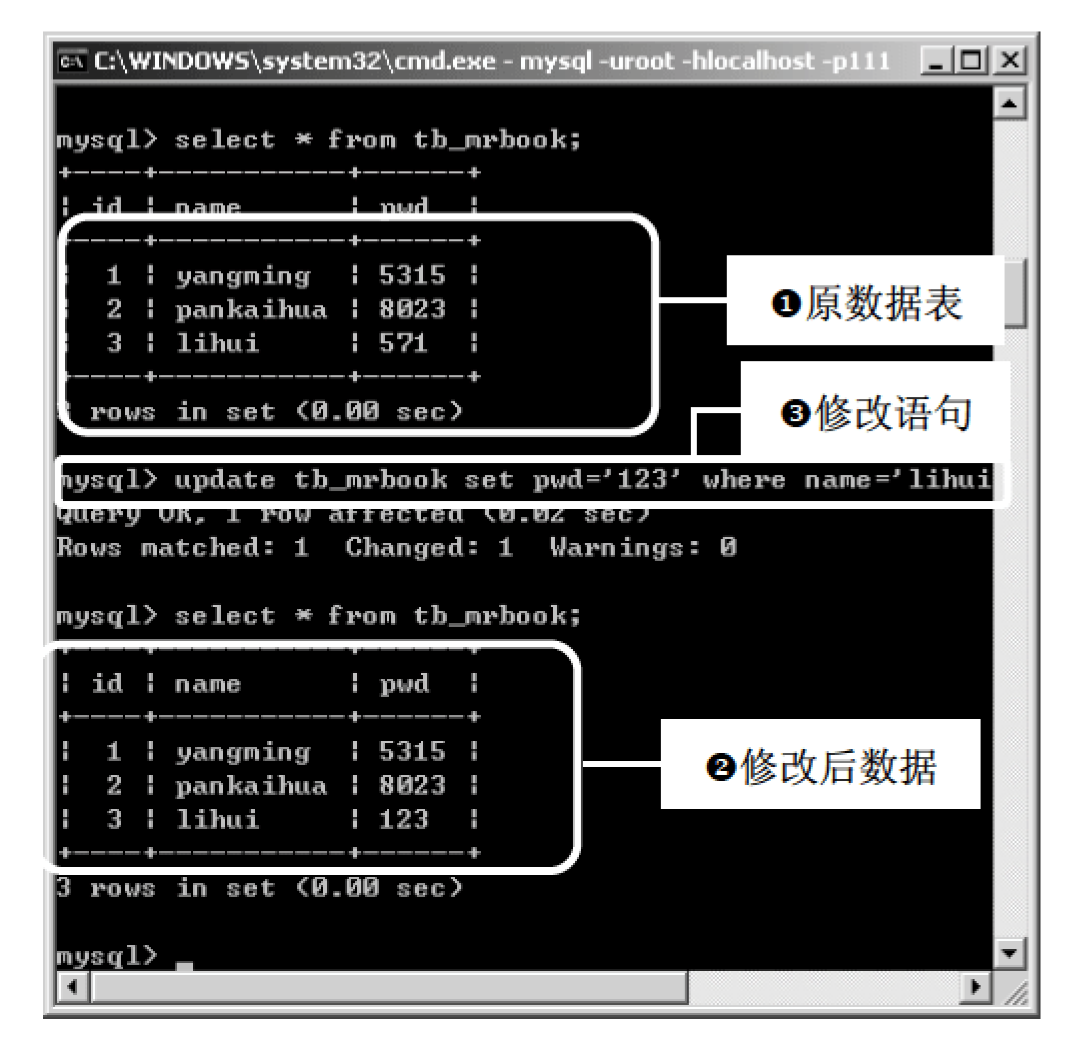



图 7.43　修改图书信息表

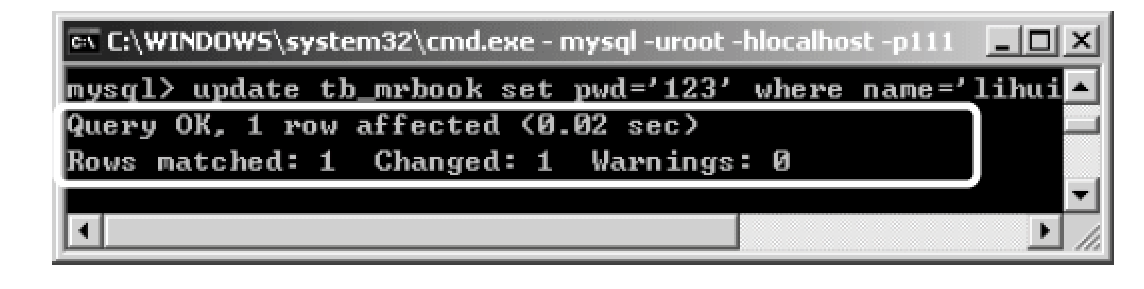


图 7.44　修改成功提示

照猫画虎：重命名 tb_mrbook 数据表。提示：rename table 数据表名 1to 数据表名 2。（20 分）

7.8.5　基本功训练 5——删除 tb_mrbook 表中指定数据

视频讲解：光盘\mr\07\lx\删除 tb_mrbook 表中指定数据.exe

删除图书信息表中所有数据或者说清空数据表在实际编程中很少见，一般都是指定删除一条或几条数据。本实例通过 delete 语句删除图书信息表中指定数据，运行结果如图 7.45 所示。



实现过程如下：

（1）开启 MySQL 服务，输入命令连接 MySQL，创建数据库和数据表，将数据表命名为 tb_mrbook，并通过 INSERT 语句向数据表中添加信息，通过 SELECT 语句查看表信息，如图 7.45 中❶原数据表所示。

（2）通过 DELETE 语句删除 id 等于 2 的数据，代码如下：

```
delete from tb_mrbook where id=2;
```

（3）当提示符窗口出现如图 7.46 所示的内容，表示数据更新成功。

（4）通过 SELECT 语句查询数据表中内容，显示结果如图 7.45 中❷删除后数据所示。

（5）断开 MySQL 服务器连接，关闭 MySQL 服务。

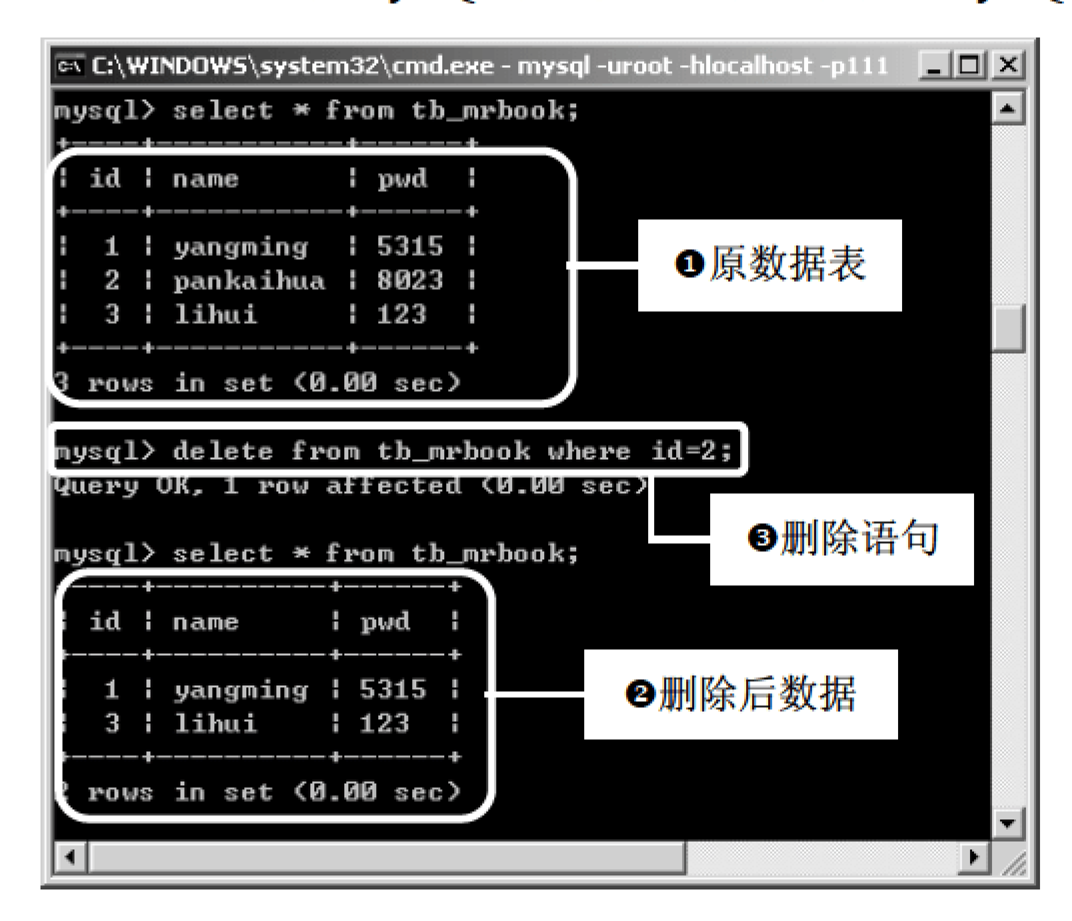


图 7.45　修改图书信息表

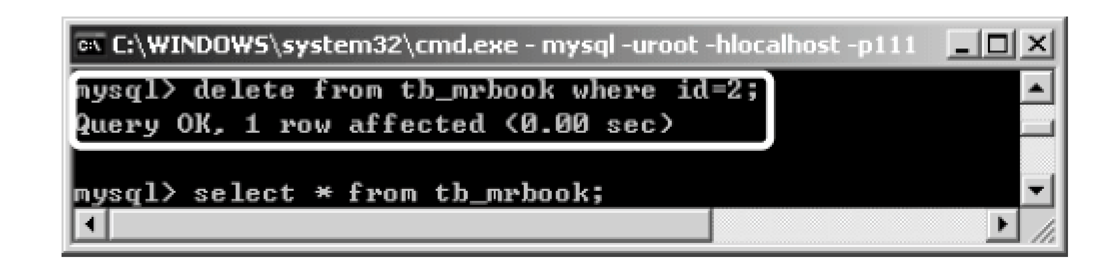


图 7.46　删除成功提示

照猫画虎：删除数据表中的所有数据。提示：应用“delete from tb_mrbook;”清除数据表中的所有信息。（20 分）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

7.9　情景应用——拓展与实践

7.9.1　情景应用 1——通过命令模式备份和恢复数据库

视频讲解：光盘\mr\07\lx\通过命令模式备份和恢复数据库.exe

1. 数据的备份

在命令模式下完成对数据的备份，使用的是 MYSQLDUMP 命令。通过该命令可以将数据以文本文件的形式存储到指定的文件夹下。通过 MYSQLDUMP 命令备份整个数据库的操作步骤如下：

（1）选择“开始”菜单中的“运行”命令，如图 7.47 所示。

（2）在弹出的如图 7.48 所示的对话框中输入“cmd”，单击“确定”按钮，进入命令模式。



图 7.47　选择“运行”命令



图 7.48　进行命令模式

（3）在命令模式中直接输入“mysqldump -uroot -p111 db_admin>E:\db_admin.txt”，然后按 Enter 键即可，如图 7.49 所示。



图 7.49　通过命令备份 db_database16 数据库

其中，-uroot 中的 root 是用户名，而-p111 中的 111 是密码，db_admin 是数据库名，E:\db_admin.txt 是数据库备份存储的位置。

说明：输入命令的过程中，在-uroot 中是没有空格的，在该命令的结尾处也没有任何的结束符，只要按 Enter 键即可。

2. 数据恢复

既然可以对数据库进行备份，那么就一定可以对数据库文件进行恢复操作。执行数据库的恢复操作使用的是 MYSQL 命令。实现数据库恢复的操作步骤如下：

（1）选择“开始”菜单中的“运行”命令。

（2）在弹出的对话框中输入“cmd”，单击“确定”按钮，进入命令模式。

（3）在命令模式中直接输入“mysql -uroot -p111 db_admin <E:\db_admin.txt”，然后按 Enter 键即可，如图 7.50 所示。

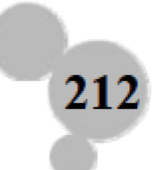


图 7.50　通过命令恢复 db_admin 数据库

其中，-uroot 中的 root 是用户名，而-proot 中的 111 是密码，db_admin 是要恢复的数据库名，E:\db_admin.txt 是数据库备份文件存储的位置。

说明：进行数据库的恢复时，在 MySQL 数据库中必须存在一个空的、将要恢复的数据库，否则在执行恢复操作时会出错。

DIY：通过手动方式完成数据库的备份与恢复。提示：通过手动方式直接备份 MySQL 数据库安装路径下 data 文件夹中的数据库文件。（20 分）



7.9.2　情景应用 2——MySQL 的存储过程

视频讲解：光盘\mr\07\lx\MySQL 的存储过程.exe

为了保证数据的完整性和一致性，提高应用的性能，常采用存储过程技术。MySQL 在 5.0 之前并不支持存储过程，随着 MySQL 技术的日趋完善，存储过程将在以后的项目中被得到广泛的应用。

一个存储过程由名字、参数列表，以及可以包括很多 SQL 语句的 SQL 语句集组成。下面为一个存储过程的定义过程。

```
create procedure proc_name (in parameter integer)
begin
declare variable varchar(20);
if parameter=1 then
set variable='MySQL';
else
set variable='PHP';
end if;
insert into tb (name) values (variable);
end;
```

MySQL 中存储过程的建立以关键字 create procedure 开始，后面紧跟存储过程的名称和参数。MySQL 的存储过程名称不区分大小写，如 PROCE1()和 proce1()代表同一存储过程名。存储过程名不能与 MySQL 数据库中的内建函数重名。

存储过程的参数一般由 3 部分组成。第一部分可以是 in、out 或 inout。in 表示向存储过程中传入参数；out 表示向外传出参数；inout 表示定义的参数可传入存储过程并可以被存储过程修改后传出存储过程，存储过程默认为传入参数，所以参数 in 可以省略。第二部分为参数名。第三部分为参数的类型，该类型为 MySQL 数据库中所有可用的字段类型，如果有多个参数，参数之间可以用逗号进行分隔。

MySQL 存储过程的语句块以 begin 开始，以 end 结束。语句体中可以包含变量的声明、控制语句、SQL 查询语句等。由于存储过程内部语句要以分号“;”结束，所以在定义存储过程前应将语句结束标志“;”更改为其他字符，并且保证该字符在存储过程中出现的几率也应较低，可以用关键字 delimiter 代替，例如：

```
MySQL>delimiter //
```

存储过程创建之后，可用如下语句进行删除，参数 proc_name 指存储过程名。

```
drop procedure proc_name
```

在 MySQL 的命令中创建存储过程的具体步骤如下：

（1）MySQL 存储过程是在命令提示符下创建的，所以首先应该打开“命令提示符”。

（2）进入“命令提示符”后，首先应该登录 MySQL 数据库服务器，在命令提示符下输入如下命令：

```
MySQL –u 用户名 –p 用户密码
```

（3）更改语句结束符号，本实例将语句结束符更改为“//”，代码如下：

```
delimiter //
```

（4）创建存储过程前应首先选择某个数据库，代码如下：

```
use 数据库名
```

（5）创建存储过程。

下面编写一个实现用户注册功能的存储过程。在创建存储过程前，为了防止结束符冲突，首先使用 delimiter 语句将 MySQL 默认的结束符“;”更改为“//”，然后创建存储过程，其创建代码如图 7.51 所示。

```
C:\>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.0.51b-community-nt-log MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use db_database09;
Database changed
mysql> delimiter //
mysql> create procedure pro_login(in un varchar(50), in pwd varchar(50))
    -> begin
    -> select * from tb_login where username=un and password = pwd;
    -> end;
    -> //
Query OK, 0 rows affected (0.06 sec)
```

图 7.51　创建用户登录验证的存储过程

（6）通过 call 语句调用存储过程。

```
call pro_login('".$username."', '".$password."')
```

DIY：利用存储过程实现多表查询。(20 分)

7.9.3　情景应用 3——使用 phpMyAdmin 设置编码格式

视频讲解：光盘\mr\07\lx\使用 phpMyAdmin 设置编码格式.exe

将页面、程序文件、数据库与数据表设置成统一的编码格式可以使程序运行时不至于出现乱码。一般情况下，设置页面的编码格式由 HTML 中的 meta 标签实现，设置程序文件的编码格式是由 header()函数实现的，设置数据库与数据表的编码格式可以通过 phpMyAdmin 实现。实现过程如下：

（1）登录到 phpMyAdmin 图形化工具页面，创建数据库名称，选择编码格式，如图 7.52 所示。



图 7.52　设置数据库编码格式

（2）创建数据表，定义数据表字段，并设置编码格式，如图 7.53 所示。





图 7.53　设置字段编码格式

DIY：通过命令模式设置编码格式。提示：通过“set names 编码格式”语句实现。(20 分)

7.9.4　情景应用 4——通过 phpMyAdmin 添加服务器新用户

视频讲解：光盘\mr\07\lx\通过 phpMyAdmin 添加服务器新用户.exe

在 phpMyAdmin 图形化管理工具中，不但可以对 MySQL 数据库进行各种操作，而且可以添加服务器的新用户，并为新添加的用户设置权限。

在 phpMyAdmin 中添加 MySQL 服务器新用户的步骤如下：

（1）单击 phpMyAdmin 主界面中的权限超链接，打开服务器用户操作界面，如图 7.54 所示。

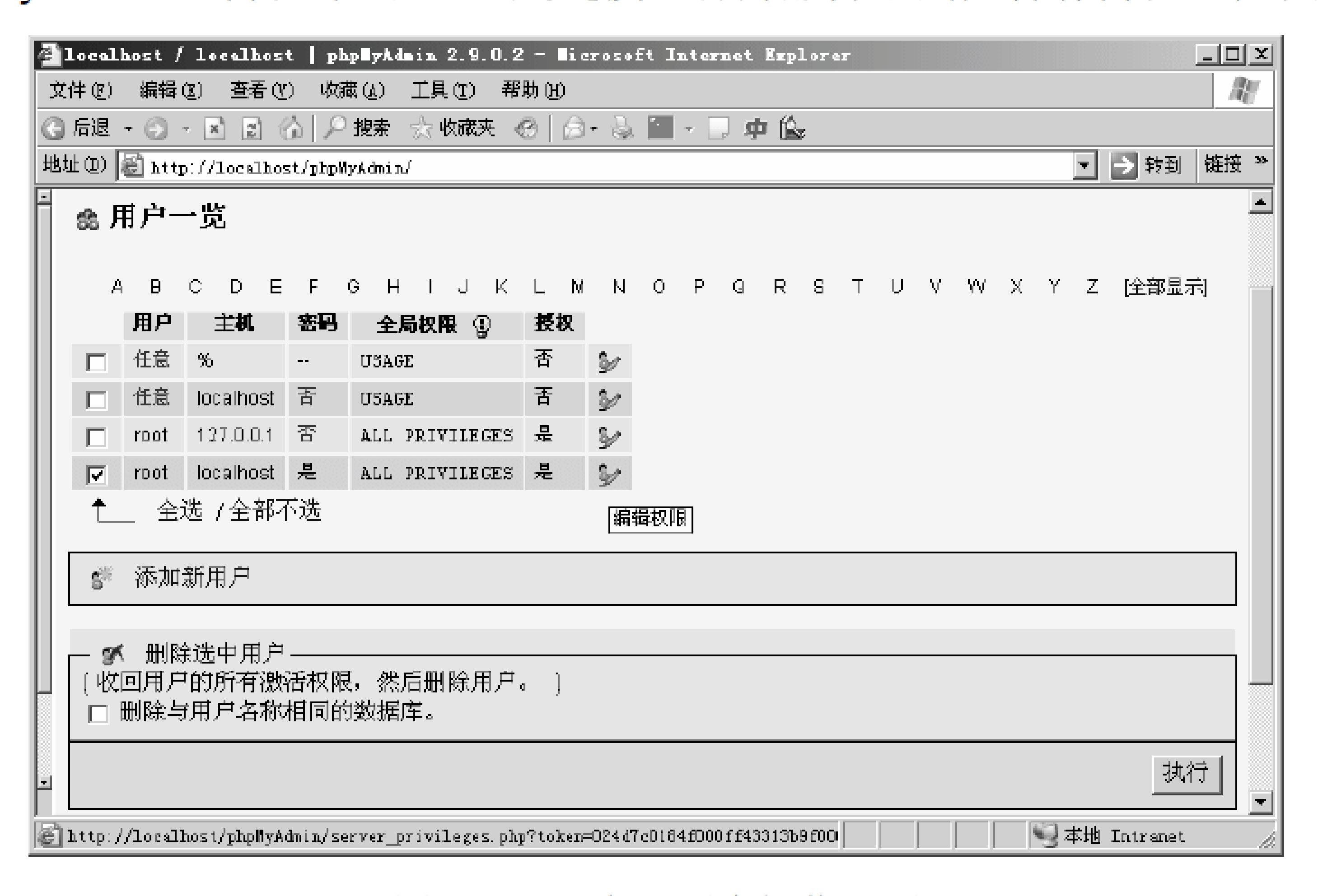


图 7.54　服务器用户操作界面

（2）在该界面中单击添加新用户。进入到如图 7.55 所示界面，在其中设置用户名、密码、主机，并对新用户的权限进行设置。设置完成后，单击“执行”按钮，完成对新用户的添加操作，返回主页面，将提

示新用户添加成功。

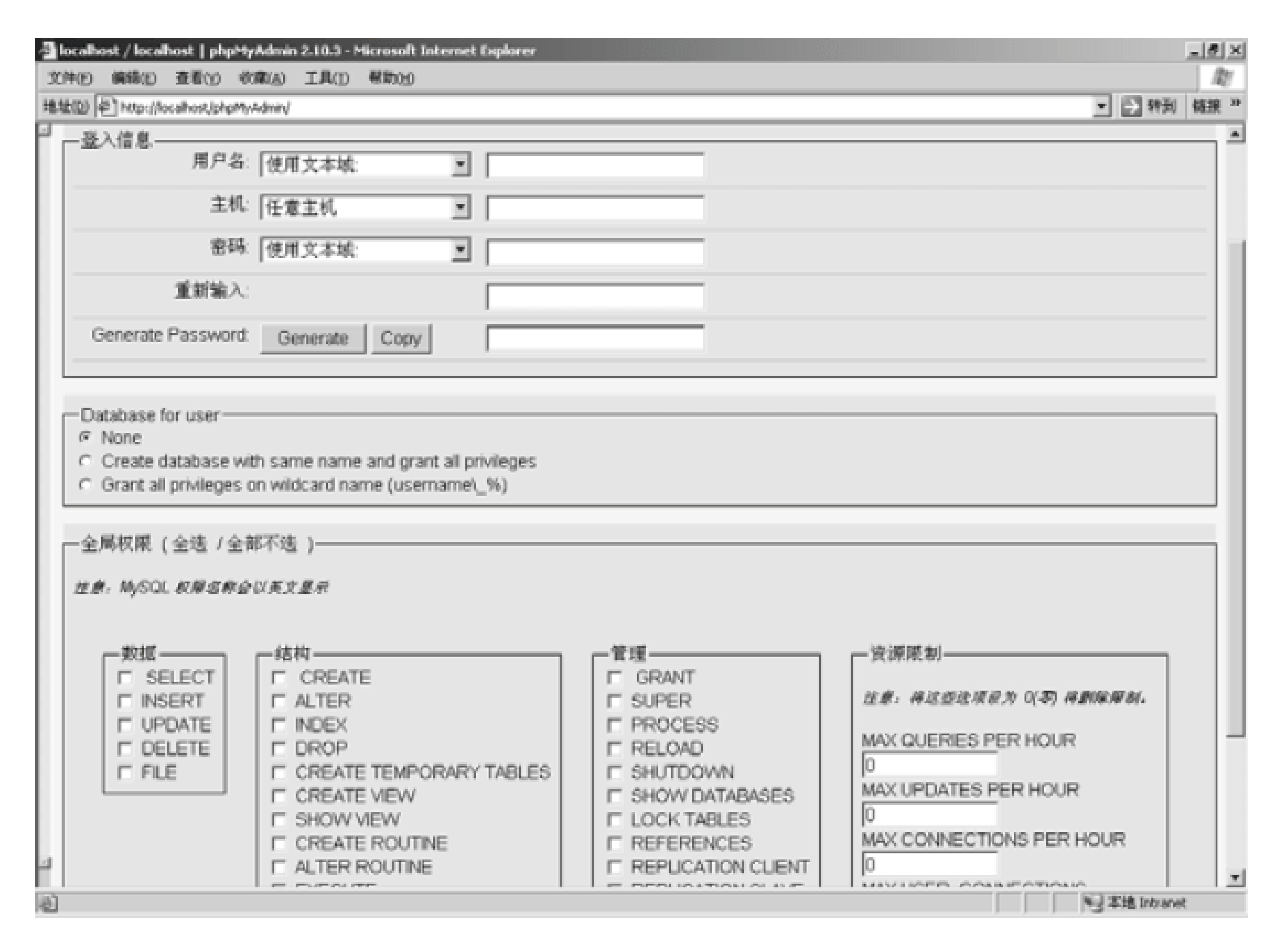


图 7.55　设置添加用户信息

DIY：删除添加的新用户。（20 分）

7.9.5　情景应用 5——在 phpMyAdmin 中重置 MySQL 服务器登录密码

视频讲解：光盘\mr\07\lx\在 phpMyAdmin 中重置 MySQL 服务器登录密码.exe

在 phpMyAdmin 图形化管理工具中，不但可以对 MySQL 数据库进行各种操作，而且可以对用户的权限进行设置，同时还可以对 MySQL 服务器的登录密码进行重置。

在 phpMyAdmin 中重置 MySQL 服务器登录密码的步骤如下：

（1）单击 phpMyAdmin 主界面中的权限超链接，打开服务器用户操作界面，如图 7.56 所示。



图 7.56　服务器用户操作界面

（2）在该界面中，可以对指定用户的权限进行编辑，可以添加新用户和删除指定的用户。这里选择指定的用户，单击（编辑权限）超链接，对指定用户的权限进行设置，进入到如图 7.57 所示界面。

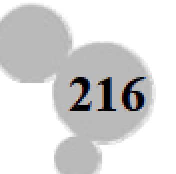




图 7.57　编辑用户权限

在图 7.57 所示的界面中，可以设置用户的权限、修改密码、更改登录用户信息和复制用户。在输入新密码和确认密码之后，单击“执行”按钮，完成对用户密码的修改操作，返回主页面，将提示密码修改成功。

DIY：设置登录用户的权限。(20 分)

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

7.10　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．关于 MySQL_select_db 的作用描述正确的是（　　）。

A．连接数据库　　B．连接并选取数据库　　C．连接并打开数据库　　D．选取数据库

2．MySQL_insert_id()函数的作用是（　　）。

A．查看下一次插入记录时的 ID 号　　B．查看刚刚插入过的自动增长 ID 值

C．查看一共做过多少次 insert 操作　　D．查看一共有多少条记录

3．修改 MySQL 用户 root 的密码的指令是（　　）。

A．MySQLadmin –u root password test　　B．MySQL –u root password test

C．MySQL –u root –p test　　D．MySQL –u root –password test

4．查询所有年龄在 20 岁以下的学生姓名及其年龄，以下正确的是（　　）。

A．SELECT Sname，Sage FROM Student WHERE Sage <=20;

B．SELECT Sname，Sage FROM Student WHERE NOT Sage <20;

C．SELECT Sname，Sage FROM Student WHERE NOT Sage >20;

D．SELECT Sname，Sage FROM Student WHERE Sage <20;

5．导出数据库正确的方法为（　　　　　　）。

A．MySQLdump 数据库名 >文件名　　　　　　B．MySQLdump 数据库名 >>文件名

C．MySQLdump 数据库名 文件名　　　　　　D．MySQLdump 数据库名 =文件名

二、填空题（每题 10 分，5 道题）

1．删除数据表中 id=5 的记录，其用到的主要代码是（　　　　　）。

2．通过命令模式设置编码格式的函数是（　　　　　）。

3．MySQL 数据库的备份和恢复，可以通过数据库的（　　　　　）和（　　　　　）来完成。

4．假设数据库 mydb 中有一个表 tb_mrbook，查询 tb_mrbook 表，按照图书价格降序排列，显示 3 条记录，其代码是（　　　　　）。

5．假设数据库 mydb 中有一个表 tb1，表中有 6 个字段，主键为 ID，有 10 条记录，ID 从 0～9，以下代码的输出结果是（　　　　　）。

```
select  *  from  tb1 where Id<=5
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

7.11　行动指南

开始日期：______年____月____日　　　　　　　　　　　　　　结束日期：______年____月____日

序号	内　　容	行动指南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）创建一个数据库 db_shop，然后查看 MySQL 服务器中所有的数据库，确认数据库 db_shop 是否创建成功。如果该数据库成功创建，则选择该数据库并进行删除操作。	
		（2）在数据库 db_shop 中创建一个 tb_shangpin 的数据表，包括 id、name、price 和 amount 字段。	
		（3）将 tb_shangpin 的数据表更名为 tb_shop，并确认更名操作是否成功。	

续表

序号	内　容	行 动 指 南
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

7.12 成功可以复制——图文世界的缔造者约翰·沃洛克

成就：桌面出版、电子文档和图形处理软件市场先驱和领导者。其开发的 PostScript 语言引起了一场桌面出版印刷革命，是世界最受尊敬的计算机软件创新者之一。

创始公司：Adobe 公司创始人，该公司目前已成为硅谷最成功的公司。

财富巅峰：亿万富翁，Adobe 公司最大的股东。

最高学历：博士

出生日期：1940 年

超级语录：“我想对孩子们说，除了我之外，爱因斯坦九年级数学考试也不及格，牛顿也是几何考试不及格。所以，考试不能说明书发明者或成功者今后的前途，学生们完全可以怀疑别人告诉他们的事情。”

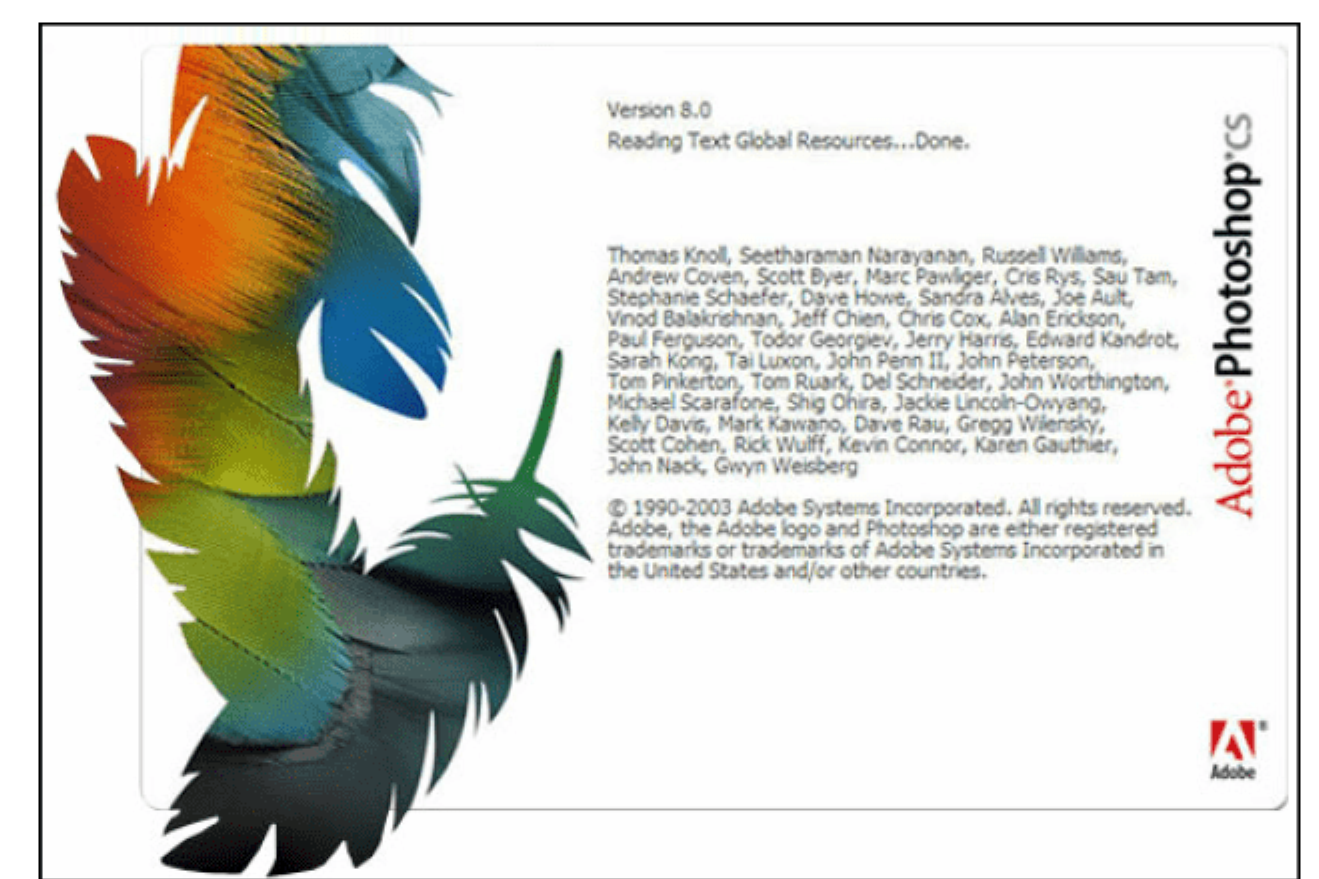
成功历程

1940 年，约翰·沃洛克（John Warnock）出生在美国犹他州的盐湖城。小时候，他并没有显示出任何过人的才华，直到读中学九年级时，代数考试仍不及格。有一次，学校组织智商测试，测试主持人询问他今后打算干什么工作？他想了想回答道：“或许，我想成为一名工程师。”主持者毫不客气地打断他的话：“测试结果表明，在工程学领域，你成功的概率几乎是零。”幸运的是，他的数学老师帮助他在数学中找到了乐趣。之后他的数学成绩稳步上升，终于如愿地考上了犹他大学，并先后取得了数学学士学位、数字硕士学位、机电工程博士学位。

1965 年，约翰·沃洛克在犹他大学获得数学硕士学位，并开始在犹他大学计算中心工作。一次，一个正在研究图形程序的学生在开发中遇到了困难，他随意询问了正在工作的约翰·沃洛克，希望能找到点灵感。没有想到，约翰·沃洛克提出了一种前所未有的解决方法，方法简单明了，但大学里所有的软件高手

都没想到。这个小插曲改变了约翰·沃洛克的生活，他变成了学校里的软件高手，这也促使他把研究课题转到了图形处理领域。

约翰·沃洛克和同事开发了一种定义三维数据库规则的程序——JAM 语言，在 JAM 的基础上，他又与同事开发出一种打印机标准 Interpress。约翰·沃洛克发现，Interpress 是一件很有发展潜力的产品，如果进行推广，将改变整个打印和出版业。他们想将 Interpress 发展成产品，但在其工作的施乐公司是不可能的，于是他们决定自己创业。



1982 年，约翰·沃洛克和 Chuck Geschke 共同创办了 Adobe 公司。如今，他已成为这个市值几百亿美元软件企业的领袖，作为桌面出版、电子文档和图形处理软件市场先驱和领导者的 Adobe 公司，不但拥有 Photoshop、PageMaker、Acrobat Reader 软件，PDF 格式和 PostScript 语言也已是世界范围的标准。2005 年，美国《财富》杂志把 Adobe 公司评为硅谷最成功的两家公司之一。

深度评价

约翰·沃洛克之所以成功，是因为他能以开发改变世界的产品作为目标，并从中得到最大满足。他说“金钱或者其他类似的东西是很好的，但这并不是他们创立 Adobe 公司的原因，金钱只是衡量经营公司的好坏程度和对市场的影响程度，开发伟大产品带来的乐趣和满足比任何其他东西给我的动力都大。”

第 8 堂课

PHP 数据库编程技术

（视频讲解：205 分钟）

PHP 所支持的数据库类型较多，在这些数据库中，MySQL 数据库与 PHP 结合最好，与 Linux 系统、Apache 服务器和 PHP 语言构成了当今主流的 LAMP 网站架构模式。并且 PHP 提供了多种操作 MySQL 数据库的方式，从而适合不同需求和不同类型项目的需要，本堂课将系统讲解如何通过 PHP 内置的 MySQL 函数库操作 MySQL 数据库。

学习摘要：

- **PHP 操作 MySQL 数据库的步骤**
- **PHP 操作 MySQL 数据库的方法**
- **向 MySQL 数据库中添加数据**
- **浏览 MySQL 数据库中的数据**
- **编辑 MySQL 数据库中的数据**
- **删除 MySQL 数据库中的数据**
- **查询 MySQL 数据库中的数据**

8.1　PHP 操作 MySQL 数据库的步骤

MySQL 是一款广受欢迎的数据库，由于它是开源的半商业软件，所以市场占有率高，备受 PHP 开发者的青睐，一直被认为是 PHP 的最好搭档。PHP 具有强大的数据库支持能力，本节主要讲解 PHP 操作 MySQL 数据库的基本思路。

PHP 操作 MySQL 数据库的步骤如图 8.1 所示。

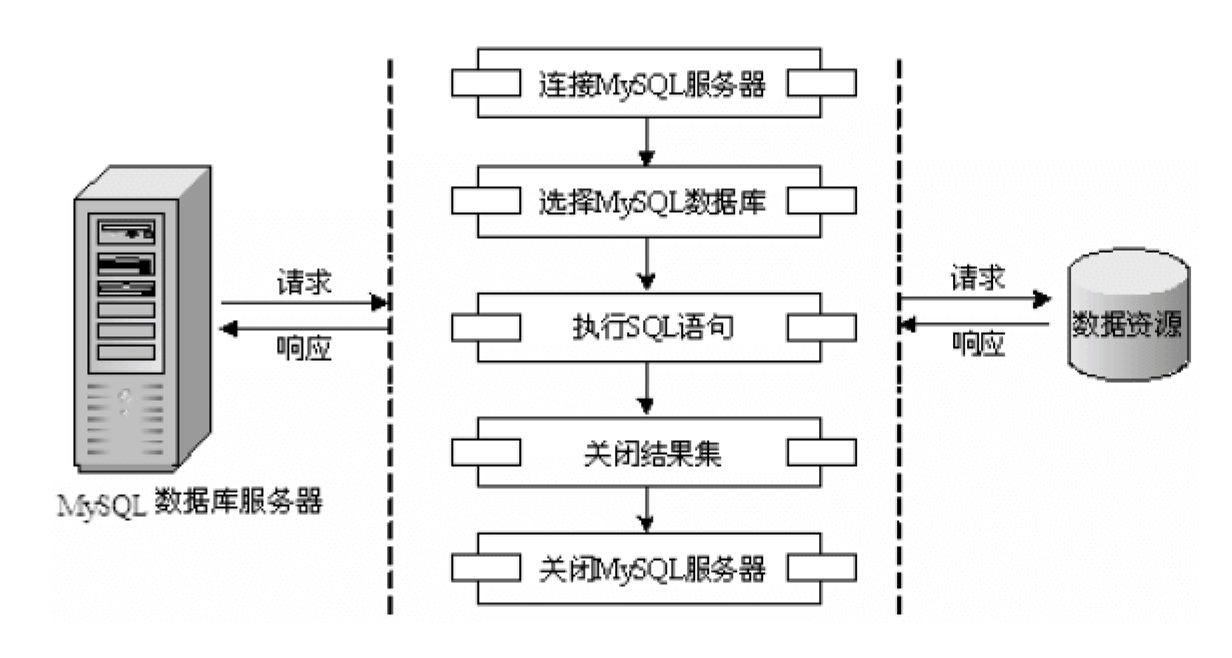


图 8.1　PHP 操作 MySQL 数据库的步骤

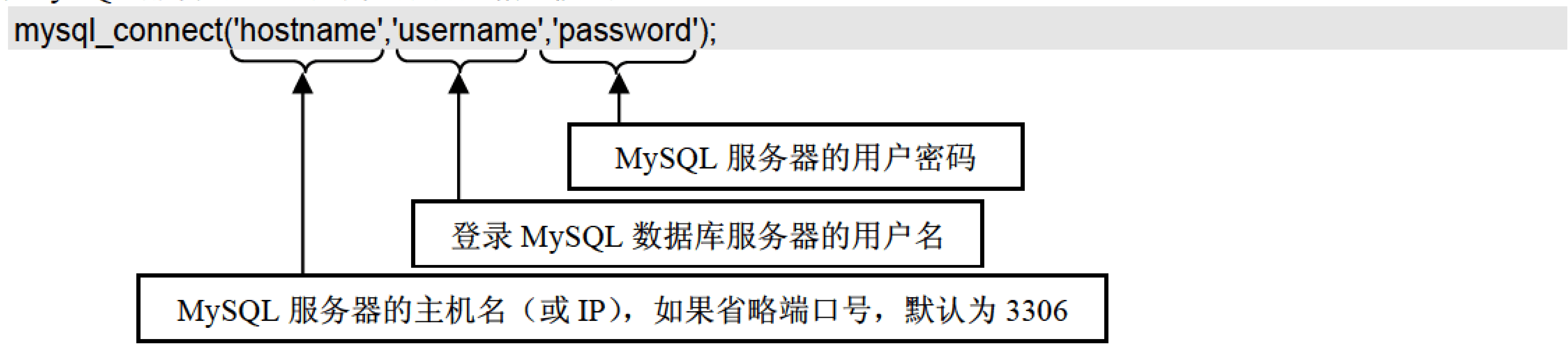
8.2　PHP 操作 MySQL 数据库的方法

PHP 中提供了很多操作 MySQL 数据库的函数，使用这些函数可以对 MySQL 数据库执行各种操作，使程序开发变得更加简单、灵活。下面介绍一些常用的 MySQL 数据库函数。

8.2.1　mysql_connect()函数连接 MySQL 服务器

要操作 MySQL 数据库，必须先与 MySQL 服务器建立连接。PHP 中提供了 mysql_connect()函数，用于连接 MySQL 服务器，该函数的语法格式如下：

```
mysql_connect('hostname','username','password');
```



该函数的返回值用于表示这个数据库连接。如果连接成功，则函数返回一个连接标识，失败则返回 FALSE。例如，使用 mysql_connect()函数连接本地 MySQL 服务器，代码如下：

```
<?php
$conn = mysql_connect("localhost", "root", "111") or die("连接数据库服务器失败！ ".mysql_error());
?>
```

为了方便查询因为连接问题而出现的错误，最好加上由 die()函数进行屏蔽的错误处理机制。使用 mysql_error()函数可以提取 MySQL 函数的错误文本，如果没有出错，则返回空字符串，如果浏览器显示



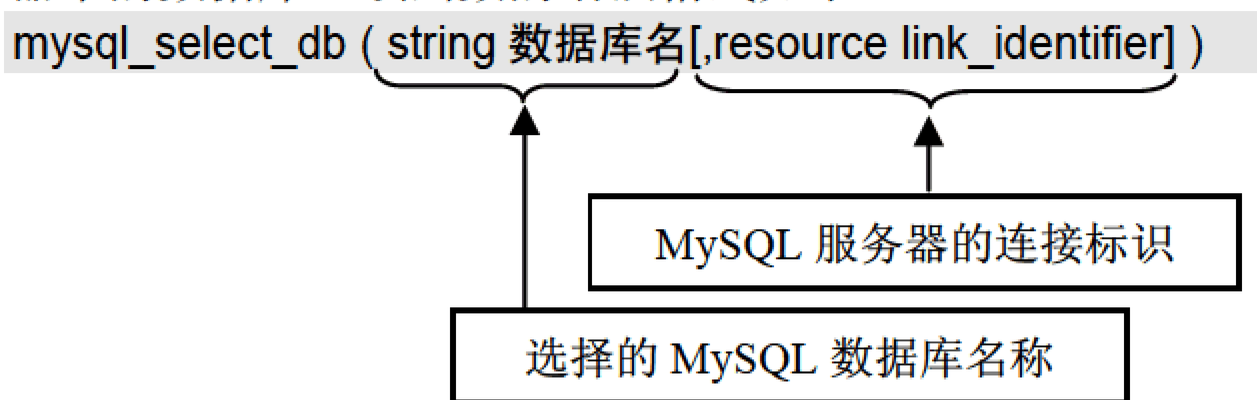
“Warning: mysql_connect()……”的字样时，说明是数据库连接的错误，这样就能迅速地发现错误位置，及时改正。

> 技巧：在 mysql_connect()函数前面添加符号“@”，用于限制这个命令的出错信息的显示。如果函数调用出错，将执行 or 后面的语句。die()函数表示向用户输出引号中的内容后，程序终止执行。这样是为了防止数据库连接出错时，用户看到一堆莫名其妙的专业名词，而是提示定制的出错信息。但在调试时不要屏蔽出错信息，避免出错后难以找到问题。

8.2.2　mysql_select_db()函数选择 MySQL 数据库

与 MySQL 服务器建立连接后，要确定所要连接的数据库，使用 mysql_select_db()函数可以连接 MySQL 服务器中的数据库，该函数的语法格式如下：

```
mysql_select_db ( string 数据库名[,resource link_identifier] )
```



例如，与本地 MySQL 服务器中的 db_database08 数据库建立连接，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接 MySQL 数据库服务器
$select=mysql_select_db("db_database08",$conn);         //连接服务器中的 db_database08 表
if($select){                                            //判断是否连接成功
    echo "数据库连接成功！";
}
?>
```

因为连接数据库的代码是正确的，所以运行结果为：数据库连接成功！

> 技巧：在开发一个完整的 Web 程序过程中，经常需要连接数据库，如果总是重复编写代码，会造成代码的冗余，而且不利于程序维护。所以通常将连接 MySQL 数据库的代码单独建立一个名为 conn.php 的文件，存储在根目录下的 conn 文件夹中，在需要使用的脚本中通过 require 语句包含这个文件即可。

8.2.3　mysql_query()函数执行 SQL 语句

在 PHP 中，通常使用 mysql_query()函数来执行对数据库操作的 SQL 语句。mysql_query()函数的语法格式如下：

```
mysql_query ( string query [, resource link_identifier] )
```

参数 query 是传入的 SQL 语句，包括插入数据（insert）、修改记录（update）、删除记录（delete）和查询记录（select）；参数 link_identifier 是 MySQL 服务器的连接标识。

例如，向会员信息表 tb_user 中插入一条会员记录，SQL 语句的代码如下：

```
$result=mysql_query("insert into tb_user values('mr','111')",$conn);
```

例如，修改会员信息 tb_user 表中的会员记录，SQL 语句的代码如下：

```
$result=mysql_query("update tb_user set name='lx' where id='01'",$conn);
```

例如，删除会员信息 tb_user 表中的一条会员记录，SQL 语句的代码如下：

```
$result=mysql_query("delete from tb_user where name='mr'",$conn);
```

例如，查询会员信息 tb_user 表中 name 字段值为 mr 的记录，SQL 语句的代码如下：

```
$result=mysql_query("select * from tb_user where name='mr'",$conn);
```

上面的 SQL 语句代码都是将结果赋给变量$result。

8.2.4 mysql_fetch_array()函数将结果集返回到数组中

使用 mysql_query()函数执行 select 语句时，将成功返回查询结果集，返回结果集后，使用 mysql_fetch_array()函数可以获取查询结果集信息，并放入到一个数组中，将其返回。其语法格式如下：

```
array mysql_fetch_array ( resource result [, int result_type] )
```

☑ result：资源类型的参数，要传入的是由 mysql_query()函数返回的数据指针。

☑ result_type：可选项，设置结果集数组的表述方式。参数值设置为 MYSQL_ASSOC，表示数组采用关联索引；设置为 MYSQL_NUM，表示数组采用数字索引；设置为 MYSQL_BOTH，同时包含关联和数字索引的数组。默认值是 MYSQL_BOTH。

例 8.01 下面获取新闻信息表 tb_news 中的新闻信息，并使用 mysql_fetch_array()函数返回结果集，然后使用 while 语句循环输出新闻标题及内容，代码如下。（**实例位置：光盘\mr\08\sl\8.01**）

```
<?php
/*连接数据库*/
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database08",$conn);
mysql_query("set names utf8");
$arr=mysql_query("select * from tb_news",$conn);
/*使用 while 语句循环 mysql_fetch_array()函数返回的数组*/
while($result=mysql_fetch_array($arr)){
?>
        <tr>
          <td height="25"><?php echo $result['name'];?><!--输出新闻标题--> </td>
          <td height="25"><?php echo $result['news'];?><!--输出新闻内容--><span class="STYLE3"> 
</span></td>
          </tr>
<?php
  }              //结束 while 循环
?>
```

运行结果如图 8.2 所示。

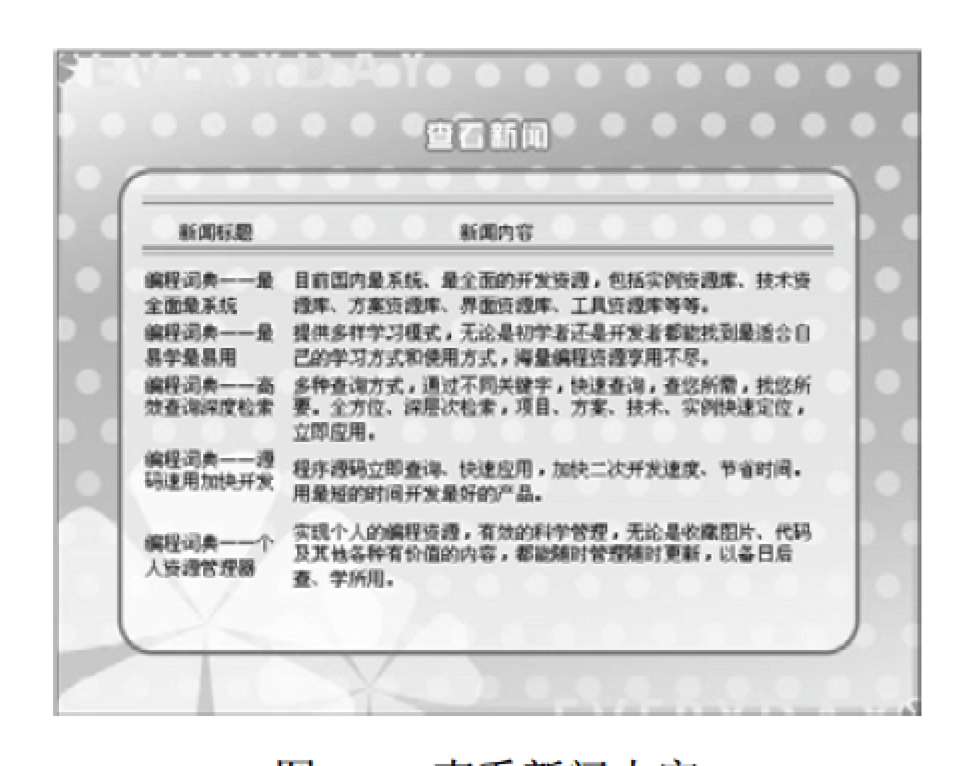


图 8.2　查看新闻内容



8.2.5　mysql_fetch_row()函数从结果集中获取一行作为枚举数组

mysql_fetch_row()函数主要实现从结果集中取得一行作为枚举数组。在应用 mysql_fetch_row()函数逐行获取结果集中的记录时，只能使用数字索引来读取数组中的数据，其语法如下：

```
array mysql_fetch_row ( resource result )
```

mysql_fetch_row()函数返回根据所取得的行生成的数组，如果没有更多行则返回 FALSE。返回数组的偏移量从 0 开始，即以$row[0]的形式访问第一个元素（只有一个元素时也是如此）。

例 8.02　本例同样是获取 db_database08 数据库中 tb_news 数据表中的新闻信息，但是与 8.2.4 节不同的是，本例应用的是 mysql_fetch_row()函数逐行获取结果集中的记录。**（实例位置：光盘\mr\08\sl\8.02）**

编写 index.php 文件。连接 db_database08 数据库，利用 mysql_query()函数执行 SQL 查询语句并返回结果集。通过 while 语句和 mysql_fetch_row()函数循环输出查询结果集，代码如下：

```
<?php
/*连接数据库*/
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database08",$conn);
mysql_query("set names utf8");
$arr=mysql_query("select * from tb_news",$conn);
/*使用 while 语句循环 mysql_fetch_row()函数返回的数组*/
while($result=mysql_fetch_row($arr)){
?>
            <tr>
              <td height="25"><?php echo $result[1];?>   <!--输出新闻标题-->   </td>
              <td height="25"><?php echo $result[2];?>   <!--输出新闻内容-->  <span  class="STYLE3"> 
</span></td>
            </tr>
<?php
  }              //结束 while 循环
?>
```

运行结果如图 8.3 所示。

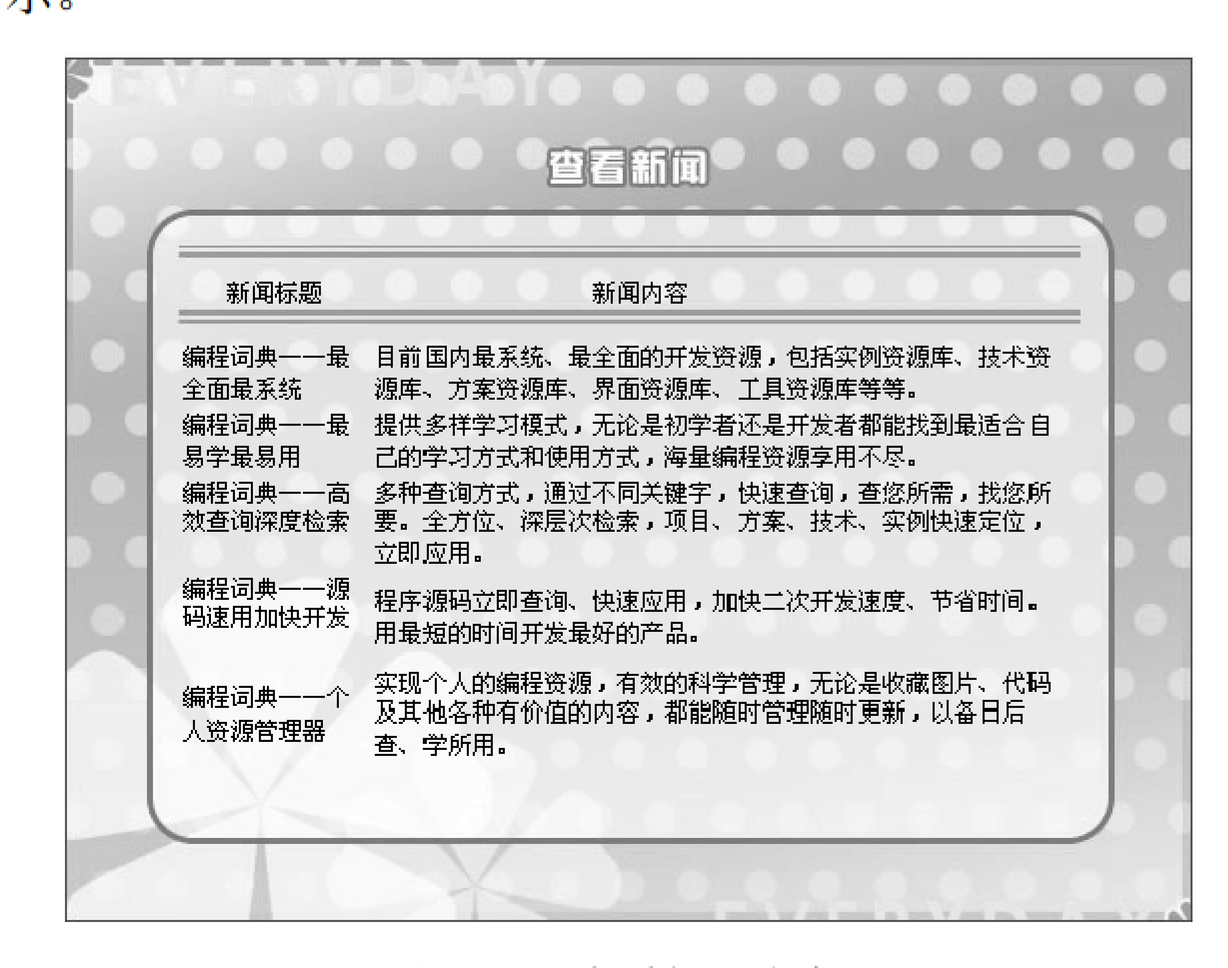


图 8.3　查看新闻内容

你问我答：mysql_fetch_array()函数和 mysql_fetch_row()函数有什么区别？

使用 mysql_fetch_array()函数获取到的数组可以是数字索引数组，也可以是关联数组，而使用 mysql_fetch_row()函数获取到的数组，只可以为数字索引数组。

8.2.6 mysql_num_rows()函数获取查询结果集中的记录数

使用 mysql_num_rows()函数可以获取由 select 语句查询到的结果集中行的数目。mysql_num_rows()函数的语法如下：

```
int mysql_num_rows ( resource result )
```

此命令仅对 SELECT 语句有效。要取得被 INSERT、UPDATE 或者 DELETE 语句所影响到的行的数目，需使用 mysql_affected_rows()函数。

例 8.03 下面使用 mysql_num_rows()函数获取查询到的新闻条数，关键代码如下。**（实例位置：光盘\mr\08\sl\8.03）**

```
<?php
/*连接数据库*/
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database08",$conn);
mysql_query("set names utf8");
$arr=mysql_query("select * from tb_news",$conn);
/*使用 while 语句循环 mysql_fetch_array()函数返回的数组*/
while($result=mysql_fetch_array($arr)){
?>
        <tr>
          <td height="25"><?php echo $result['name'];?><!--输出新闻标题--> </td>
          <td height="25"><?php echo $result['news'];?><!--输出新闻内容--><span class="STYLE3"> 
</span></td>
          </tr>
<?php
  }                //结束 while 循环
?>
<?php echo mysql_num_rows($arr);?> 条                //查询新闻总数
```

运行结果如图 8.4 所示。

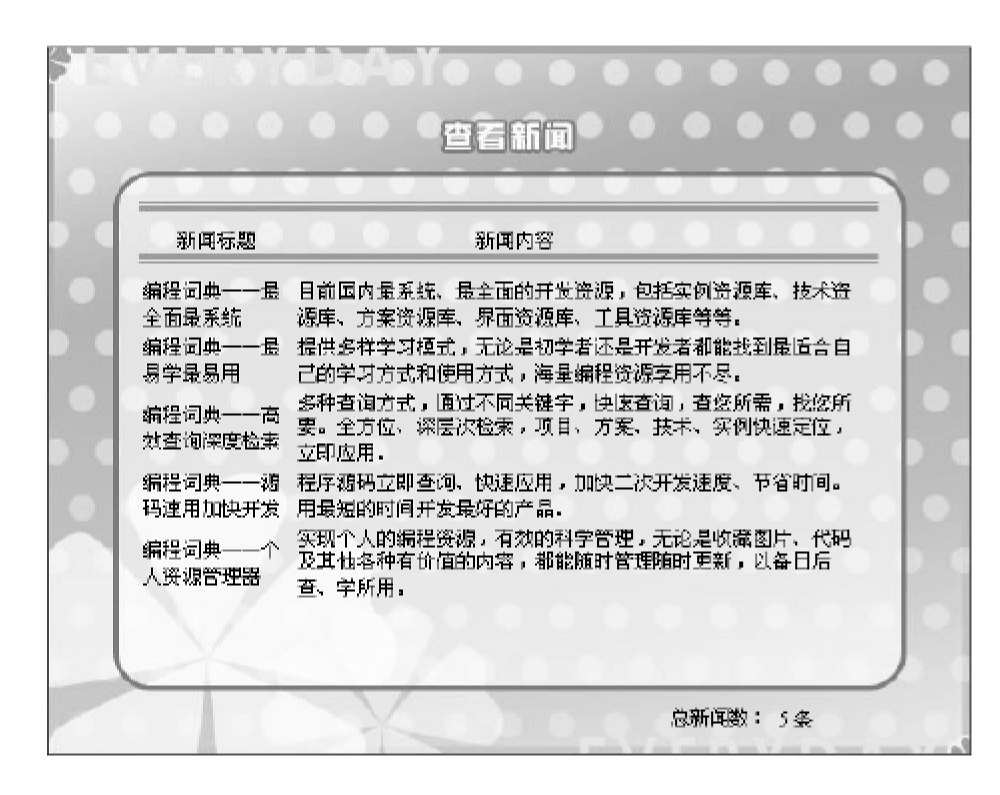


图 8.4 获取查询到的新闻数



8.2.7　mysql_free_result()函数释放内存

mysql_free_result()函数用于释放内存，数据库操作完成后，需要关闭结果集，以释放系统资源，该函数的语法格式如下：

```
mysql_free_result($result);
```

mysql_free_result()函数将释放所有与结果标识符 result 所关联的内存。该函数仅需要在考虑到返回很大的结果集时会占用多少内存时调用。在脚本结束后所有关联的内存都会被自动释放，下面举一个使用该函数释放内存的例子，具体代码如下：

```
<?php
mysql_connect("localhost", "root", "111");
mysql_select_db("db_mrbook");
$result = mysql_query("select * from tb_book");
while($row=mysql_fetch_array($result)) {
    printf ("ID: %s    BookName: %s",$row[0],$row["bookname"])."<br>";
}
mysql_free_result($result);        //释放结果内存空间
?>
```

本示例的运行结果如下：

```
ID: 2 BookName: PHP 程序开发范例宝典 ID: 3 BookName: PHP+MySQL 数据库系统开发完全手册 ID: 4 BookName: PHP+MySQL 数据库系统开发完全手册 ID: 11 BookName: PHP 程序开发范例宝典
```

技巧： 如果在多个网页中都要频繁进行数据库访问，则可以建立与数据库服务器的持续连接来提高效率。因为每次与数据库服务器连接需要较长的时间和较大的资源开销，持续的连接相对来说会更有效。建立持续连接的方法就是在数据库连接时，调用函数 mysql_pconnect()代替 mysql_connect()函数。建立的持续连接在本程序结束时，不需要调用 mysql_close()函数来关闭。下次程序再次执行 mysql_pconnect()函数时，系统自动直接返回已经建立的持续连接的 ID 号，而不再去真的连接数据库。

8.2.8　mysql_close()函数关闭连接

每使用一次 mysql_connect()或 mysql_query()函数，都会消耗系统资源。在少量用户访问 Web 网站时问题还不大，但如果用户连接超过一定数量时，就会造成系统性能的下降，甚至死机。为了避免这种现象的发生，在完成数据库的操作后，应使用 mysql_close()函数关闭与 MySQL 服务器的连接，以节省系统资源。mysql_close()函数的语法如下：

```
mysql_close($conn);
```

在 Web 网站的实际项目开发过程中，经常需要在 Web 页面中查询数据信息。查询后使用 mssql_close()函数关闭数据源。

例 8.04　在浏览新闻信息页面中，浏览信息后应用 mysql_close()函数关闭已经连接的数据源，代码如下。**（实例位置：光盘\mr\08\sl\8.04）**

```
<?php
/*连接数据库*/
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database08",$conn);
```

```
mysql_query("set names utf8");
$arr=mysql_query("select * from tb_news",$conn);
/*使用 while 语句循环 mysql_fetch_array()函数返回的数组*/
while($result=mysql_fetch_array($arr)){
?>
            <tr>
              <td height="25"><?php echo $result['name'];?>    <!--输出新闻标题-->   </td>
              <td height="25"><?php echo $result['news'];?>    <!--输出新闻内容-->  <span class="STYLE3">
  </span></td>
            </tr>
<?php
  }              //结束 while 循环
mysql_close($conn);                    //关闭数据源
?>
```

技巧：PHP 中与数据库的连接是非持久连接，系统会自动回收，一般不用设置关闭。但如果一次性返回的结果集比较大，或网站访问量比较多，则最好使用 mysql_close()函数手动进行释放。

8.3 管理 MySQL 数据库中的数据

管理 MySQL 数据库中的数据主要是对数据进行添加、修改、删除、查询等操作，只有熟练地掌握这部分知识，才能够独立开发出基于 PHP 的数据库应用。

8.3.1 向数据库中添加数据

向数据库中添加数据主要通过 mysql_query()函数和 insert 语句实现。

例 8.05 本例实现发表博客文章，填写文章标题及文章内容，当用户单击“提交”按钮时，判断文章标题及内容是否为空，如果不为空则将数据添加到数据库中，关键代码如下。（**实例位置：光盘\mr\08\sl\8.05**）

```
<?php
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database08",$conn);
mysql_query("set names utf8");
if(isset($_POST['Submit']) and $_POST['textfield']!=null and $_POST['textarea']!=null and $_POST['Submit'] ==
"提交"){
     $insert=mysql_query("insert into tb_news(name,news) values('".$_POST['textfield']."','".$_POST['textarea']."')");
     if($insert){
     echo "<script> alert('发表成功!'); window.location.href='index.php'</script>";
     }else{
     echo "<script> alert('发表失败!'); window.location.href='index.php'</script>";
     }
}else{
     echo "<script> alert('发表失败!'); window.location.href='index.php'</script>";
}
?>
```

运行结果如图 8.5 所示。



图 8.5　添加文章

8.3.2　浏览数据库中的数据

浏览数据库中的数据主要通过 mysql_query()函数和 select 语句查询数据，使用 mysql_fetch_assoc()函数将查询结果返回到数组中。

例 8.06　下面浏览 tb_news 表中的新闻信息，具体代码如下。（**实例位置：光盘\mr\08\sl\8.06**）

```
<?php
/*连接数据库*/
$conn=mysql_connect("localhost","root","111");
mysql_select_db("db_database08",$conn);
mysql_query("set names utf8");
$arr=mysql_query("select * from tb_news",$conn);
/*使用 while 语句循环 mysql_fetch_assoc()函数返回的数组*/
while($result=mysql_fetch_assoc($arr)){
?>
            <tr>
              <td height="25"><?php echo $result['name'];?>   <!--输出新闻标题-->   </td>
              <td height="25"><?php echo $result['news'];?>   <!--输出新闻内容-->  <span
class="STYLE3"> </span> </td>
            </tr>
<?php
  }              //结束 while 循环
?>
```

运行结果如图 8.6 所示。

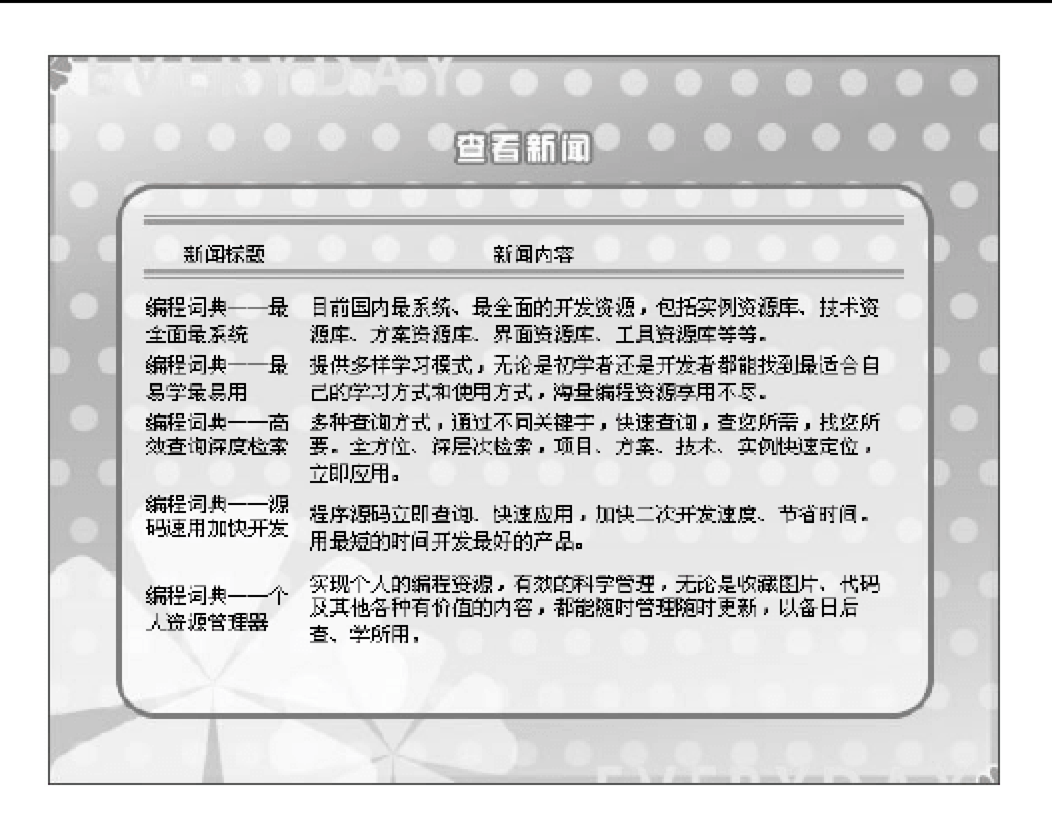


图 8.6　浏览新闻信息

8.3.3　编辑数据库中的数据

编辑数据主要通过 mysql_query()函数和 update 语句实现。

例 8.07　下面编辑新闻信息表中的新闻信息。（**实例位置：光盘\mr\08\sl\8.07**）

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");                    //连接数据库服务器
mysql_select_db("db_database08",$conn);                           //连接 db_database08 数据库
mysql_query("set names utf8");                                    //设置数据库编码格式
?>
```

（2）创建 index.php 文件，显示所有新闻信息，代码如下：

```
<?php
include("conn.php");                                              //包含 conn.php 文件
$arr=mysql_query("select * from tb_news",$conn);                  //查询数据
/*使用 while 语句循环 mysql_fetch_array()函数返回的数组*/
while($result=mysql_fetch_array($arr)){
?>
        <tr>
          <td height="25"><?php echo $result['name'];?><!--输出新闻标题--> </td>
          <td><?php echo $result['news'];?> <!--输出新闻内容--> <span class="STYLE3"> </span> </td>
          <td><label>
            <input type="hidden" name="id" value="<?php echo $result['id'];?>" />
            <div align="center">
            <a href="update.php?id=<?php echo $result['id'];?>">编辑</a></div>
          </label></td>
          </tr>
<?php
  }              //结束 while 循环
?>
```

（3）创建 update.php 文件，显示要编辑的新闻内容，代码如下：

```
<form id="form1" name="form1" method="post" action="update_ok.php">
<?php
include("conn.php");        //包含 conn.php 文件
```

```
	$arr=mysql_query("select * from tb_news where id='".$_GET['id']."'",$conn);		//定义查询语句
	$select=mysql_fetch_array($arr);						//循环输出查询内容
?>
		<input name="name" type="text" size="40" value="<?php echo $select['name'];?>"/>
		<textarea name="news" cols="40" rows="10"><?php echo $select['news'];?></textarea>
		<input type="submit" name="Submit" value="保存" />
		<input type="hidden" name="id" value="<?php echo $select['id'];?>" />
</form>
```

（4）创建 update_ok.php 文件，完成新闻信息的编辑操作，代码如下：

```
<?php
include("conn.php");				//包含 conn.php 文件
if(isset($_POST['id']) and isset($_POST['Submit']) and $_POST['Submit']=="保存"){
	$update=mysql_query("update tb_news set name='".$_POST['name']."',news='".$_POST['news']."' where
id='".$_POST['id']."'",$conn);
	if($update){
		echo	"<script> alert('修改成功!'); window.location.href='index.php'</script>";
	}else{
		echo	"<script> alert('修改失败!'); window.location.href='index.php'</script>";
	}
}
?>
```

运行结果如图 8.7 所示。

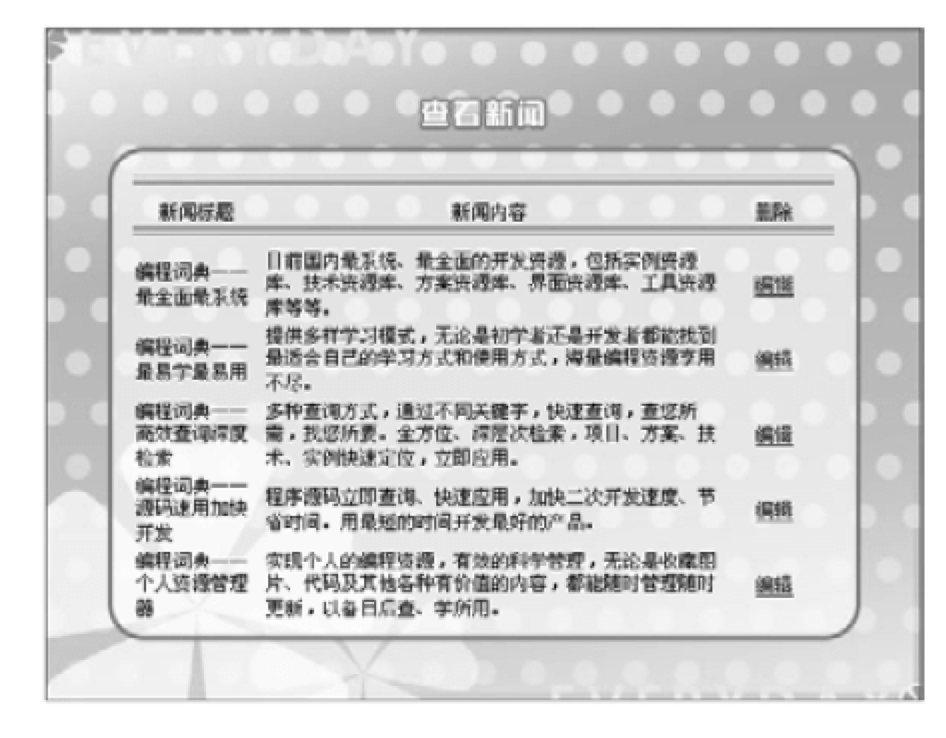


图 8.7　编辑新闻信息

8.3.4　从数据库中删除数据

数据的删除应用 delete 语句，而在 PHP 中需要通过 mysql_query()函数来执行 delete 删除语句，完成 MySQL 数据库中数据的删除操作。

例 8.08　下面删除新闻信息表中的新闻信息。（**实例位置：光盘\mr\08\sl\8.08**）

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");		//连接数据库服务器
mysql_select_db("db_database08",$conn);			//连接 db_database08 数据库
mysql_query("set names utf8");				//设置数据库编码格式
?>
```

（2）创建 index.php 文件，显示所有新闻信息，代码如下：

```
<?php
include("conn.php");                                    //包含 conn.php 文件
$arr=mysql_query("select * from tb_news",$conn);        //查询数据
/*使用 while 语句循环 mysql_fetch_array()函数返回的数组*/
while($result=mysql_fetch_array($arr)){
?>
        <tr>
          <td height="25"><?php echo $result['name'];?>    <!--输出新闻标题--> </td>
          <td><?php echo $result['news'];?>                <!--输出新闻内容-->  <span class="STYLE3">
 </span></td>
          <td><label>
            <input type="hidden" name="id" value="<?php echo $result['id'];?>" />
            <div align="center">
              <input name="Submit" type="submit" class="STYLE3" value="删除" />
              </div>
          </label></td>
          </tr>
<?php
  }            //结束 while 循环
?>
```

（3）在 index_ok.php 文件中根据超链接传递的 ID 值，完成删除新闻信息操作，代码如下：

```
<?php
include("conn.php");                                                //包含 conn.php 文件
if(isset($_POST['id']) and isset($_POST['Submit']) and $_POST['Submit']=="删除"){
     $delete=mysql_query("delete from tb_news where id='".$_POST['id']."'",$conn);
     if($delete){
          echo   "<script> alert('删除成功!'); window.location.href='index.php'</script>";
     }else{
          echo   "<script> alert('删除失败!'); window.location.href='index.php'</script>";
     }
}
?>
```

执行程序，运行结果如图 8.8 所示。

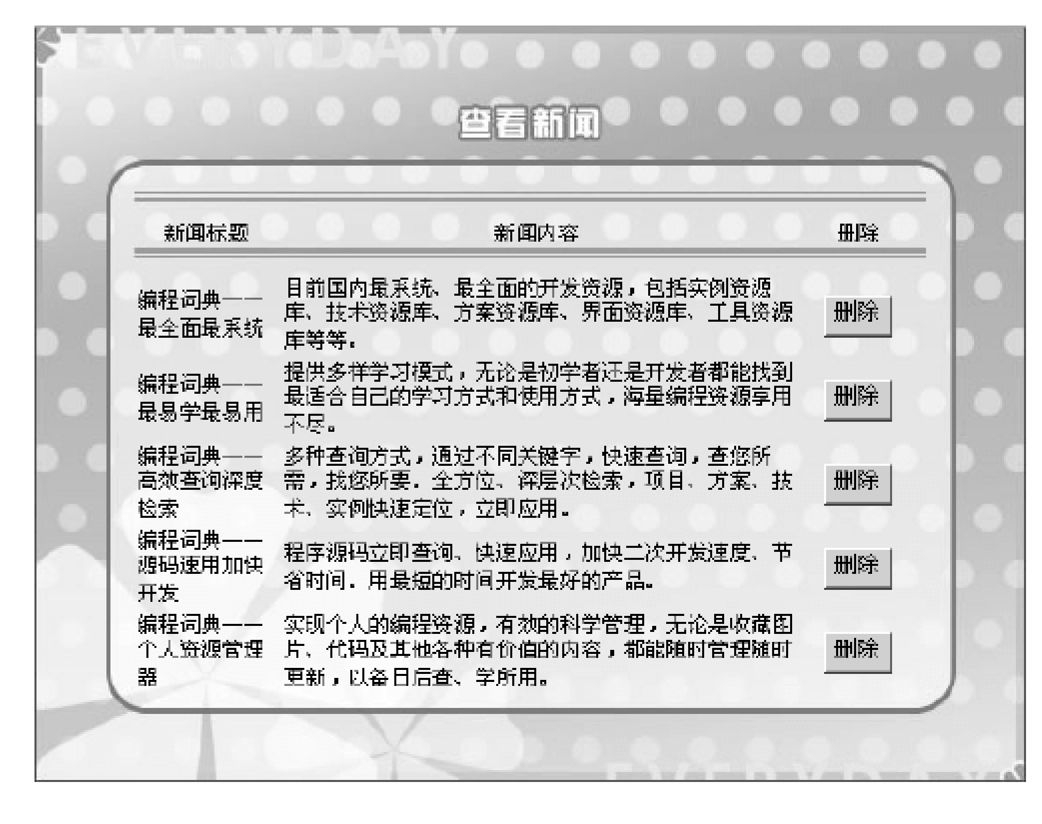


图 8.8　删除新闻信息

8.3.5　批量数据操作

对数据库中的数据进行管理过程中，如果要删除的数据非常多，执行单条删除数据的操作就不适合了，这时应该使用批量删除数据来实现数据库中信息的删除。通过数据的批量删除可以快速地删除多条数据，以减少操作执行的时间。

例 8.09　下面批量清理新闻信息表中陈旧的新闻信息。（**实例位置：光盘\mr\08\sl\8.09**）

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");              //连接数据库服务器
mysql_select_db("db_database08",$conn);                     //连接 db_database08 数据库
mysql_query("set names utf8");                              //设置数据库编码格式
?>
```

（2）创建 index.php 文件，显示所有新闻信息，代码如下：

```
<?php
include("conn.php");
$arr=mysql_query("select * from tb_news",$conn);            //查询数据
/*使用 while 语句循环 mysql_fetch_array()函数返回的数组*/
while($result=mysql_fetch_array($arr)){
?>
          <tr>
            <td><label>
              <label>
              <input type="checkbox" name="checkbox[]" value="<?php echo $result['id'];?>" />
              </label>
            </label></td>
            <td height="25"><?php echo $result['name'];?>    <!--输出新闻标题-->   </td>
            <td><?php echo $result['news'];?>                <!--输出新闻内容-->  <span class="STYLE3">  
</span></td>
          </tr>
<?php
  }              //结束 while 循环
?>
```

（3）创建 index_ok.php 页面，完成批量删除操作，代码如下：

```
<?php
include("conn.php");                                        //包含 conn.php 文件
if(isset($_POST['Submit']) and $_POST['Submit']=="删除" and $_POST['checkbox']!=""){
                                                            //判断是否执行删除操作
    for($i=0;$i<count($_POST['checkbox']);$i++){            //遍历复选框获取到的新闻 id 序号
        $sql=mysql_query("delete from tb_news where id='".$_POST['checkbox'][$i]."'",$conn);
                                                            //执行删除操作
    }
    if($sql){
        echo  "<script> alert('删除成功!'); window.location.href='index.php'</script>";
    }else{
        echo  "<script> alert('删除失败!'); window.location.href='index.php'</script>";
    }
```

```
    }else{
        echo  "<script> alert('请选择要删除的内容!'); window.location.href='index.php'</script>";
    }
?>
```

执行程序，运行结果如图 8.9 所示。

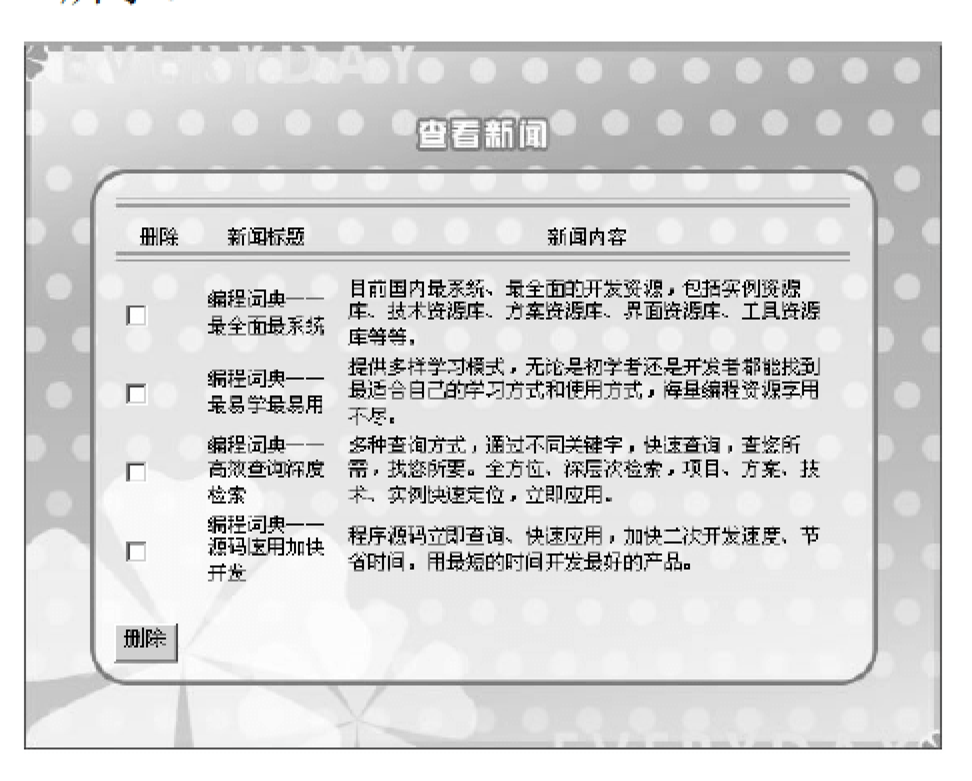


图 8.9　批量删除数据信息

8.4　照猫画虎——基本功训练

8.4.1　基本功训练 1——连接 db_database08 数据库

视频讲解：光盘\mr\08\lx\连接 db_database08 数据库.exe

实例位置：光盘\mr\08\zmhh\01

下面连接 db_database08 数据库，如果连接成功则进入 index_ok.php 页，如图 8.10 所示；否则跳转到 index_no.php 页，运行结果如图 8.11 所示。



图 8.10　连接成功页面



图 8.11　连接失败页面

本例主要通过 mysql_connect()函数连接数据库服务器，通过 mysql_select_db()函数连接数据，具体代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");                    //连接数据库服务器
$select=mysql_select_db("db_database08",$conn);                   //连接数据库
mysql_query("set names utf8");
if($select){
    echo "<script>alert('连接成功！ '); window.location.href='index_ok.php'</script>";
}else{
```



```
    echo "<script>alert('连接失败！'); window.location.href='index_no.php'</script>";
}
?>
```

照猫画虎：根据上面的例子，数据库测试完成后关闭数据源。提示：使用 mysql_close()函数可以关闭数据源。（10 分）（实例位置：光盘\mr\08\zmhh\01_zmhh）

8.4.2　基本功训练 2——查询图书信息表中的前 3 条记录

视频讲解：光盘\mr\08\lx\查询图书信息表中的前 3 条记录.exe

实例位置：光盘\mr\08\zmhh\02

查询图书信息表 tb_book 中的前 3 条图书信息记录，单击“查询”按钮，即可查询前 3 条图书信息记录，运行结果如图 8.12 所示。



图 8.12　查询前 3 条图书信息记录

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");              //连接数据库服务器
mysql_select_db("db_database08",$conn);                      //连接 db_database08 数据库
mysql_query("set names utf8");                               //设置数据库编码格式
?>
```

（2）创建 index.php 文件，设计图书信息查询表单页面，效果如图 8.12 所示。

（3）完成查询操作，代码如下：

```
<?php
include("conn.php");            //包含 conn.php 文件
if(isset($_POST['Submit']) and $_POST['Submit']=="查询"){                 //判断用户是否执行查询操作
    $select=mysql_query("select * from tb_book where id limit 3 ",$conn);  //查询数据库中前 3 条记录
} else{
    $select=mysql_query("select * from tb_book",$conn);                   //查询所有记录
}
?>
<?php
    while($rows=mysql_fetch_array($select)){                              //使用 while 循环查询结果
```

```
?>
        <tr>
          <td><?php echo $rows['id'];?> </td>                    <!--输出 id-->
          <td><?php echo $rows['name'];?> </td>                  <!--输出书名-->
          <td><?php echo $rows['pen'];?> </td>                   <!--输出作者-->
          <td><?php echo $rows['pub'];?> </td>                   <!--输出出版社-->
          <td><?php echo $rows['date'];?> </td>                  <!--输出出版时间-->
          <td><?php echo $rows['rate'];?> </td>                  <!--输出价格-->
        </tr>
<?php
    }                                                                 //结束循环
?>
```

照猫画虎：根据上面的例子，查询图书信息表中出版社为“清华大学出版社”的图书信息。（20 分）（实例位置：光盘\mr\08\zmhh\02_zmhh）

8.4.3 基本功训练 3——对查询结果进行降序排列输出

视频讲解：光盘\mr\08\lx\对查询结果进行降序排列输出.exe

实例位置：光盘\mr\08\zmhh\03

对图书信息表 tb_news 中的图书信息记录按图书序号进行降序排列，单击“降序排列”按钮，即可按图书序号进行降序排列，运行结果如图 8.13 所示。



图 8.13　将图书信息按 ID 进行降序排列

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_database08",$conn);                 //连接 db_database08 数据库
mysql_query("set names utf8");                          //设置数据库编码格式
?>
```

（2）创建 index.php 文件，设计图书信息查询表单页面，效果如图 8.13 所示。

（3）完成降序排列操作，代码如下：



```
<?php
include("conn.php");                                          //包含 conn.php 文件
if(!isset($_POST['Submit2'])){                                //判断按钮事件为空时
      $select=mysql_query("select * from tb_book",$conn);     //显示数据表中所有记录
      $rows=mysql_fetch_array($select);
}else{                                                        //否则按 id 进行降序排列，并获取返回值
      $select=mysql_query("select * from tb_book order by id desc",$conn);
      $rows=mysql_fetch_array($select);
}
?>

<?php
      do{                                                     //使用 do…while 循环
?>
       <tr>
             <td><?php echo $rows['id'];?> </td>         <!--输出 id-->
             <td><?php echo $rows['name'];?> </td>       <!--输出书名-->
             <td><?php echo $rows['pen'];?> </td>        <!--输出作者-->
             <td><?php echo $rows['pub'];?> </td>        <!--输出出版社-->
             <td><?php echo $rows['date'];?> </td>       <!--输出出版时间-->
             <td><?php echo $rows['rate'];?> </td>       <!--输出价格-->
          </tr>
<?php
      }while($rows=mysql_fetch_array($select));               //执行降序排列操作
?>
```

照猫画虎：对上例中的图书信息按出版时间的先后顺序进行降序排列。（20 分）（实例位置：光盘\mr\08\zmhh\03_zmhh）

8.4.4　基本功训练 4——更新指定图书的信息

视频讲解：光盘\mr\08\lx\更新指定图书的信息.exe

实例位置：光盘\mr\08\zmhh\04

更新指定的图书信息，当用户单击“修改”超链接时，显示图书详细信息，并可对图书信息进行编辑，然后单击“提交”按钮即可完成修改操作，运行结果如图 8.14 所示。



图 8.14　更新指定的图书信息

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");            //连接数据库服务器
mysql_select_db("db_database08",$conn);                   //连接 db_database08 数据库
mysql_query("set names utf8");                            //设置数据库编码格式
?>
```

（2）创建 index.php 文件，设计图书信息浏览页面，显示所有图书信息，代码如下：

```
<?php
include("conn.php");                                      //包含 conn.php 文件
    $select=mysql_query("select * from tb_book");         //查询数据
    while($row=mysql_fetch_array($select)){               //循环输出查询内容
?>
    <tr>
        <td><?php echo $row['id'];?> </td>           <!--输出 id-->
        <td><?php echo $row['name'];?> </td>         <!--输出书名-->
        <td><?php echo $row['pen'];?> </td>          <!--输出作者-->
        <td><?php echo $row['pub'];?> </td>          <!--输出出版社-->
        <td><?php echo $row['date'];?> </td>         <!--输出出版时间-->
        <td><?php echo $row['rate'];?> </td>         <!--输出价格-->
        <td><div align="center"><a href="update.php?id=<?php echo $row['id'];?>">修改</a>
          <input type="hidden" name="id" value="<?php echo $rows['id'];?>"/>
        </div></td>
    </tr>
<?php
    }
?>
```

（3）创建 update.php 文件，设计编辑图书信息页面，代码如下：

```
<?php
include("conn.php");                                                          //包含 conn.php 文件
    $select=mysql_query("select * from tb_book where id='".$_GET['id']."'");  //定义查询语句
    $row=mysql_fetch_array($select);                                          //循环输出查询内容
?>
<form id="form1" name="form1" method="post" action="update_ok.php">
<input name="name" type="text" size="40" value="<?php echo $row['name'];?>"/>
<input name="pen" type="text" size="40" value="<?php echo $row['pen'];?>"/>
<input name="pub" type="text" size="40" value="<?php echo $row['pub'];?>"/>
<input type="text" name="date" value="<?php echo $row['date'];?>"/>
<input type="text" name="rate" value="<?php echo $row['rate'];?>"/>
            <input type="submit" name="Submit" value="提交" />
            <input type="submit" name="Submit2" value="取消" />
            <input type="hidden" name="id" value="<?php echo $row['id'];?>" />
</form>
```

（4）创建 update_ok.php 文件，完成图书信息修改操作，代码如下：

```
<?php
include("conn.php");
if(isset($_POST['Submit']) and $_POST['Submit']=="提交" and $_POST['name']!="" and $_POST['pen']!="" and
$_POST['pub']!=""){
    $update=mysql_query("update tb_book set name='".$_POST['name']."',pen='".$_POST['pen']."',pub='".$_
POST['pub']."',date='".$_POST['date']."',rate='".$_POST['rate']."' where id='".$_POST['id']."'",$conn);
```

```
        if($update){
            echo "<script>alert('修改成功！');window.location.href='index.php'</script>";
        }else{
            echo "<script>alert('修改失败！');window.location.href='index.php'</script>";
        }
}else{
        echo "<script>alert('信息不能为空！');window.location.href='index.php'</script>";
}
?>
```

照猫画虎：在上例的浏览图书信息页面中统计图书记录总条数。提示：使用 mysql_num_rows()函数可以实现该功能。(20 分)(实例位置：光盘\mr\08\zmhh\04_zmhh)

8.4.5　基本功训练 5——删除指定的用户信息

视频讲解：光盘\mr\08\lx\删除指定的用户信息.exe

实例位置：光盘\mr\08\zmhh\05

删除指定用户信息，当用户单击“删除”超链接时，即可删除本条用户信息，运行结果如图 8.15 所示。



图 8.15　删除用户信息

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_database08",$conn);                 //连接 db_database08 数据库
mysql_query("set names utf8");                          //设置数据库编码格式
?>
```

（2）创建 index.php 页面，显示所有用户信息，代码如下：

```
<?php
include("conn.php");                                    //包含 conn.php 文件
    $select=mysql_query("select * from tb_user");       //查询数据
    while($row=mysql_fetch_array($select)){             //循环输出查询内容
?>
        <tr>
```

```
                <td height="36"><?php echo $row['name'];?> </td>
                <td><?php echo $row['pwd'];?> </td>
                <td><?php echo $row['sex'];?> </td>
                <td><?php echo $row['email'];?> </td>
                <td><a href="update.php?id=<?php echo $row['id'];?>">修改</a>/删除
                <input type="hidden" name="id" value="<?php echo $row['id'];?>"/></td>
                                                                <!--定义隐藏域，获取 id-->
            </tr>
<?php
      }                                                                 //结束循环
?>
```

（3）创建 delete.php 页面，完成删除操作，代码如下：

```
<?php
include("conn.php");                                        //包含 conn.php 文件
if(isset($_GET['id'])){                                     //判断 id 是否存在
$delete=mysql_query("delete from tb_user where id='".$_GET['id']."'",$conn);
      if($delete){
            echo "<script>alert('删除成功！');window.location.href='index.php'</script>";
      }else{
            echo "<script>alert('删除失败！');window.location.href='index.php'</script>";
      }
}
?>
```

照猫画虎：根据上面的例子，实现删除数据前给予提示，确认是否删除该条记录。提示：应用 window 对象中的 confirm()方法。（30 分）（实例位置：光盘\mr\08\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

8.5 情景应用——拓展与实践

8.5.1 情景应用 1——用户注册模块

视频讲解：光盘\mr\08\lx\用户注册模块.exe

实例位置：光盘\mr\08\qjyy\01

下面制作一个简单的用户注册页面，输入用户注册信息，单击“注册”按钮即可完成注册操作，运行结果如图 8.16 所示。

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");              //连接数据库服务器
mysql_select_db("db_database08",$conn);                     //连接 db_database08 数据库
mysql_query("set names utf8");                              //设置数据库编码格式
?>
```



图 8.16　用户注册模块

（2）创建 index.php 文件，设计用户登录页面，效果如图 8.16 所示，并使用 JavaScript 脚本技术判断用户注册信息是否为空，如果为空则弹出提示，代码如下：

```
<script language="javascript">
function checkit(){                                    //自定义函数
      if(form1.name.value==""){                        //判断用户名是否为空
               alert("请输入用户名!");
                form1.name.select();
                 return false;
           }
         if(form1.pwd.value==""){                      //判断密码是否为空
                 alert("请输入密码!");
                 form1.pwd.select();
                 return false ;
            }
            if(form1.email.value==""){                 //判断 email 是否为空
                 alert("请输入 email 地址!");
                 form1.email.select();
                 return false;
            }
                       return true ;
}
</script>
```

（3）创建 index_ok.php 文件，完成用户注册操作，代码如下：

```
<?php
include("conn.php");                                            //包含 conn.php 文件
if(isset($_POST['name']) && $_POST['name']!=""){                //判断用户名是否存在
$sql=("insert   into   tb_user(name,pwd,sex,email)   values('".$_POST['name']."','".$_POST['pwd']."','".$_POST
['sex']."','".$_POST['email']."')");
$result=mysql_query($sql,$conn);                                //执行添加操作
      if($result){                                              //判断是否执行成功
      echo   "<script> alert('注册成功!'); window.location.href='index.php'</script>";
      }else{
      echo   "<script> alert('注册失败!'); window.location.href='index.php'</script>";
      }
}
?>
```

DIY：根据上面的实例，注册时将密码进行加密，然后保存到数据库中。提示：通过 md5()函数可实现密码加密功能。(15 分)(实例位置：光盘\mr\08\qjyy\01_diy)

8.5.2 情景应用 2——用户登录模块

视频讲解：光盘\mr\08\lx\用户登录模块.exe

实例位置：光盘\mr\08\qjyy\02

下面制作一个简单的用户登录模块，当用户输入正确的用户名（mr）和密码（123456）时，即可成功登录，运行结果如图 8.17 所示。



图 8.17 用户登录模块

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");              //连接数据库服务器
mysql_select_db("db_database08",$conn);                     //连接 db_database08 数据库
mysql_query("set names utf8");                              //设置数据库编码格式
?>
```

（2）创建 index.php 文件，设计用户登录页面，效果如图 8.17 所示。

（3）创建 index_ok.php 文件，完成用户登录操作，代码如下：

```
<?php
include("conn.php");                //包含 conn.php 文件
$select=mysql_query("select * from tb_user where name='".$_POST['name']."' and pwd='".$_POST['pwd']."'",
$conn);
if(isset($_POST['Submit']) and $_POST['name']!=null and $_POST['pwd']!=null){
        if(mysql_num_rows($select)==1){
                echo "<script> alert('登录成功!'); window.location.href='index.php'</script>";
        }else{
                echo   "<script> alert('登录失败!'); window.location.href='index.php'</script>";
        }
}
?>
```

DIY：根据上面的例子，当用户登录成功后，提示用户登录名称、登录时间及登录 IP。(20 分)(实例位置：光盘\mr\08\qjyy\02_diy)

8.5.3 情景应用 3——浏览注册用户信息

视频讲解：光盘\mr\08\lx\浏览注册用户信息.exe

实例位置：光盘\mr\08\qjyy\03

下面制作一个用户信息浏览页面，在页面中显示所有用户注册信息，运行结果如图 8.18 所示。

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");              //连接数据库服务器
mysql_select_db("db_database08",$conn);                     //连接 db_database08 数据库
```

```
mysql_query("set names utf8");                    //设置数据库编码格式
?>
```



图 8.18　浏览注册用户信息

（2）创建 index.php 文件，设计浏览用户注册信息页面，效果如图 8.18 所示。

（3）在 index.php 文件中，完成浏览用户注册信息操作，代码如下：

```
<?php
include("conn.php");                                   //包含 conn.php 文件
$select=mysql_query("select * from tb_user",$conn);    //执行查询操作，并返回查询结果
?>

<?php
    while($rows=mysql_fetch_array($select)){           //使用 while 循环输出查询结果
?>
        <tr>
          <td><div align="center"><?php echo $rows['name'];?></div></td>
          <td><div align="center"><?php echo $rows['sex'];?></div></td>
          <td><div align="center"><?php echo $rows['email'];?></div></td>
        </tr>
<?php
    }                                                  //结束 while 循环
?>
```

DIY：使用 mysql_fetch_assoc()函数浏览上例中的用户注册信息。（15 分）（实例位置：光盘\mr\08\qjyy\03_diy）

8.5.4　情景应用 4——查询注册用户

视频讲解：光盘\mr\08\lx\查询注册用户.exe

实例位置：光盘\mr\08\qjyy\04

下面制作一个查询注册用户信息页面，在“请输入用户名”文本框中输入用户名，单击“查询”按钮，即可显示查询结果。运行结果如图 8.19 所示。

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");         //连接数据库服务器
mysql_select_db("db_database08",$conn);               //连接 db_database08 数据库
mysql_query("set names utf8");                         //设置数据库编码格式
?>
```



图 8.19　查询注册用户信息

（2）创建 index.php 文件，设计查询页面，效果如图 8.19 所示。

（3）在 index.php 文件中完成查询操作，代码如下：

```
<?php
include("conn.php");                                                    //包含 conn.php 文件
if(isset($_POST['Submit']) and isset($_POST['name']) and  $_POST['name']!=null and $_POST['Submit']=="查询"){    //判断查询操作
      $select="select * from tb_user where name='".$_POST['name']."'";    //定义查询语句
      $rows=mysql_query($select,$conn);                                 //执行查询
      while($row=mysql_fetch_array($rows)){                             //循环输出查询内容
?>
         <tr>
                <td height="36"><?php echo $row['name'];?> </td>
                <td><?php echo $row['pwd'];?> </td>
                <td><?php echo $row['sex'];?> </td>
                <td><?php echo $row['email'];?> </td>
             </tr>
<?php
      }
}
?>
```

DIY：根据上面的例子，实现对用户名模糊查询功能。提示：通过 like 关键字实现模糊查询，例如“name like '%".$_POST['name']."%'”，对用户名进行模糊查询。（20 分）（实例位置：光盘\mr\08\qjyy\04_diy）

8.5.5　情景应用 5——修改指定用户的注册信息

视频讲解：光盘\mr\08\lx\修改指定用户的注册信息.exe

实例位置：光盘\mr\08\qjyy\05

下面制作一个修改用户信息页面，单击“修改”超链接，即可进入用户信息修改页面，修改信息后单击“修改”按钮，即可完成修改操作，运行结果如图 8.20 所示。

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_database08",$conn);                 //连接 db_database08 数据库
mysql_query("set names utf8");                          //设置数据库编码格式
?>
```



图 8.20　用户信息修改页面

（2）创建 index.php 页面，显示所有用户信息，代码如下：

```
<?php
include("conn.php");                                    //包含 conn.php 文件
    $select=mysql_query("select * from tb_user");       //查询数据
    while($row=mysql_fetch_array($select)){              //循环输出查询内容
?>
        <tr>
            <td height="36"><?php echo $row['name'];?> </td>
            <td><?php echo $row['pwd'];?> </td>
            <td><?php echo $row['sex'];?> </td>
            <td><?php echo $row['email'];?> </td>
            <td><a href="update.php?id=<?php echo $row['id'];?>">修改</a>/删除
            <input type="hidden" name="id" value="<?php echo $row['id'];?>"/></td>
                                                                <!--定义隐藏域，获取 id-->
          </tr>
<?php
    }                                                           //结束循环
?>
```

（3）创建 update.php 页面，显示用户详细信息，代码如下：

```
<form id="form1" name="form1" method="post" action="">
<?php
include("conn.php");                                                        //包含 conn.php 文件
    $select=mysql_query("select * from tb_user where id='".$_GET['id']."'");   //定义查询语句
    $row=mysql_fetch_array($select);                                        //循环输出查询内容
?>
<input type="text" name="name" value="<?php echo $row['name'];?>"/>
<input type="text" name="pwd" value="<?php echo $row['pwd'];?>"/>
<input name="email" type="text" size="30" value="<?php echo $row['email'];?>"/>
<input type="submit" name="Submit" value="修改" />
                <input type="submit" name="Submit2" value="取消" />
                  <input type="hidden" name="id" value="<?php echo $row['id'];?>"/>
</form>
```

（4）创建 update_ok.php 页面，完成修改操作，代码如下：

```
<?php
include("conn.php");                                                              //包含 conn.php 文件
if($_POST['Submit']=="修改" and $_POST['name']!=null and $_POST['pwd']!=null){   //判断修改事件
$update=mysql_query("update  tb_user  set  name='".$_POST['name']."',pwd='".md5($_POST['pwd'])."',email
='".$_POST['email']."' where id='".$_POST['id']."'",$conn);
    if($update){
```

```
            echo "<script>alert('修改成功！');window.location.href='index.php'</script>";
        }else{
            echo "<script>alert('修改失败！');window.location.href='update.php'</script>";
        }
}else{
        echo "<script>alert('信息不能为空！');window.location.href='index.php'</script>";
}
?>
```

DIY：根据上面的实例，在修改页面中限制用户名被修改，用户名为不可更改项。（15 分）（实例位置：光盘\mr\08\qjyy\05_diy）

8.5.6 情景应用 6——分页浏览用户注册信息

视频讲解：光盘\mr\08\lx\分页浏览用户注册信息.exe

实例位置：光盘\mr\08\qjyy\06

下面在浏览用户信息页面中实现分页功能，每页显示 4 条记录。运行结果如图 8.21 所示。

具体步骤如下：

（1）创建数据库连接文件 conn.php，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_database08",$conn);                 //连接 db_database08 数据库
mysql_query("set names utf8");                          //设置数据库编码格式
?>
```



图 8.21　分页浏览用户信息

（2）创建 index.php 页面，完成分页操作，代码如下：

```
<?php
include("conn.php");
if (isset($_GET['page'])){
        $page=$_GET['page'];
}else{
        $page=1;
}
?>
<?php
    if($page){
     $page_size=4;                                       //每页显示 4 条记录
```

```
        $query="select * from tb_user";                    //定义查询语句
         $result=mysql_query($query);                      //执行查询操作
         $message_count=mysql_num_rows($result);           //获取查询总数
        $page_count=ceil($message_count/$page_size);       //获取总的页数
        $offset=($page-1)*$page_size;                      //计算每页的起始行数
        $query="select * from tb_user where id order by id desc limit $offset, $page_size";
                                                           //定义 SQL 语句，查询当前页显示的记录
         $result=mysql_query($query);                      //执行查询操作
         }
?>
<?php
            while ($myrow=mysql_fetch_array($result)){     //使用 while 循环输出查询结果
?>
            <tr>
              <td><div align="center"><?php echo $myrow['name'];?></div></td>
              <td><div align="center"><?php echo $myrow['sex'];?></div></td>
              <td><div align="center"><?php echo $myrow['email'];?></div></td>
            </tr>
<?php
       }                                                   //结束 while 循环
?>
页次：<?php echo $page;?>/<?php echo $page_count;?>页 记录：<?php echo $message_count;?>
       //输出总页数、当前页及总记录数
<?php
if($page!=1){
       echo "<a href=index.php?page=1>首页</a>";
       echo "<a href=index.php?page=".($page-1).">上一页</a>";
}
if($page<$page_count){
       echo " <a href=index.php?page=".($page+1).">下一页</a>";
       echo " <a href=index.php?page=$page_count;>尾页</a>";
}
?>
```

DIY：根据上面的实例，在页面中输出总记录数、总页数及当前页。（15 分）（实例位置：光盘\mr\08\qjyy\06_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	6	总分数	
分数								

8.6　自我测试

一、选择题（每题 10 分，5 道题）

1．连接 MySQL 数据库服务器的函数是（　　）。

A．mysql_connect()　　B．mysql_query()　　C．mysql_error()　　D．mysql_select_db

2．获取查询结果记录数使用（　　）函数。

A．mysql_fetch_array()　B．mysql_fetch_rows()　C．mysql_fetch_row()　D．mysql_num_rows()

3．查询数据表中的数据使用（　　）函数。

A．mysql_query()　B．mysql_error()　C．mysql_connect()　D．mysql_fetch_array()

4．执行数据修改操作，使用（　　）语句。

A．select　B．update　C．delete　D．insert

5．mysql_fetch_row() 和 mysql_fetch_array 的区别是（　　）。

A．mysql_fetch_row()只可以使用数字作为索引，而 mysql_fetch_array()既可以使用数字作为索引，也可以使用字符串作为索引来获取数据。

B．mysql_fetch_array()只可以使用数字作为索引，而 mysql_fetch_row()既可以使用数字作为索引，也可以使用字符串作为索引来获取数据。

二、填空题（每题 10 分，5 道题）

1．下面的代码是查询 tb_user 表中用户名字段（name）与用户输入的文本框（name）是否相同，请补充完整。

```
mysql_query("select * from tb_user where__________,$conn);
```

2．将下面连接数据库的代码补充完整。

```
$conn=____________("localhost","root","111");
_______("db_database08",_______);
_______("set names utf8");
```

3．通过 mysql_query 函数执行搜索语句获取的结果集的函数是____________。

4．查询数据表中前 4 条记录，数据表的主键为 id，请将下面的查询语句补充完整。

```
mysql_query("select * from tb_user where ________",$conn);
```

5．向数据库中添加一条用户信息记录，用户名为 mr，密码为 123456，请将下面的代码补充完整。

```
mysql_query("insert into tb_user(name,pwd) values(___)",$conn);
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

8.7　行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。

续表

序号	内　　容	行 动 指 南	
1	自我测试栏目 分数（　　）	分数>75 分	优秀，有成为编程高手的潜质。
		分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）做一个简单的单条数据录入，并将数据保存到数据库中。	
		（2）做一个简单的数据更新模块，要求更新前弹出对话框提示，是否确认更新。	
		（3）做一个批量删除数据模块，并给出提示对话框，提示本次删除数据条数。	
		（4）做一个简单的清空数据表中所有数据的模块。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

8.8　成功可以复制——射击游戏之父约翰·卡马克

成就：享誉世界的著名程序员，第一人称射击游戏之父，在电视游戏领域被尊为偶像，30 岁步入美国互动艺术和科学学院名人堂。

创始公司：id Software（世界著名的游戏公司和 3D 引擎开发公司）

财富巅峰：亿万富翁，占有 id Software 公司的 40%股份。

最高学历：高中

出生日期：1970 年 8 月 20 日

超级语录：在信息时代，客观障碍已经不复存在，所谓障碍都是主观上的。如果你想动手开发什么，你不需要几百万美元的资金，你只需要在冰箱里面放满比萨和可乐，再有一台便宜的计算机，和为之献身的决心。我们在地板上睡过，我们从河水中趟过。

成功历程

出生于德克萨斯州的约翰·卡马克（John Carmack）与许多电脑天才一样，从小就对电脑和程序设计充满狂热，七年级时的卡马克对《创世纪》和《巫术》等游戏痴迷不已，开始制作基于 Apple II 的小游戏。由于对编程的热爱，卡马克中途放弃了在密苏里大学就读计算机科学专业的机会，开始了自己艰辛的创业之路。为了获取 Softdisk 公司的程序员职位，他在雪地中徒步走了三英里才得以赶到招聘地点。在这里，约

翰·卡马克结识了约翰·罗梅洛（John Romero《文明》系列游戏的缔造者）、汤姆·豪尔（Tom Hall）等意气相投的好友，随后他们一同返回德克萨斯建立了自己的游戏公司——id Software。

约翰·卡马克是一个天才的程序设计师，仅仅依靠自学和钻研就掌握了高深的程序设计技巧，他甚至相信可以用编程完成一切。在当今这个 3D 技术飞速发展的时代，没有哪一家公司可以像 id Software 那样引领技术潮流，也没有哪一家公司可以让硬件制造商俯首称臣，这都要归功于约翰·卡马克开发的 3D 加速技术。《雷神之锤》刚刚问世时，3D 加速卡在人们眼里还只是一个可笑的空想而已，只有约翰·卡马克对 3D 技术的威力深信不疑，他为《雷神之锤》制作了一个专门在 Verite 显卡上运行的特别版本，画面看上去非常漂亮，可惜的是 Verite 显卡未能在市场上站稳脚跟。随后卡马克又采用 OpenGL 标准为《雷神之锤》制作了一个新的版本，使所有具备 3D 加速能力的显卡都能以更快的速度、更高的分辨率渲染出更华丽的图像。一时间，所有的电脑用户都争相购买这款游戏，人们甚至为了能玩上 3D 游戏而去购买昂贵的 PC，这给 id Software 和约翰·卡马克带来了上亿美元的商业利润。直到今天，显卡生产商在研发新产品之前还会先同约翰·卡马克商量一下，以确保他们的硬件可以完美地支持 id Software 出品的游戏，并确保硬件性能符合时代之潮流。



现在，约翰·卡马克仍然担当着 id Software 的游戏软件首席程序员，出自他手的电脑游戏已达数十部之多。但卡马克并不满足于现状，他的最新目标是有关火箭技术的研究项目，并为此特意成立了 Armadillo 航空宇宙技术研究公司，为着又一个梦想继续努力。

亲自开发的游戏：《德军总部 3D》（Wolfenstein 3D）、《毁灭战士》（Doom）和《雷神之锤》（Quake）等十多部经典游戏。

使用其游戏引擎开发的游戏：《反恐精英》和《荣誉勋章》等。

深度评价

如果你问比尔·盖茨，“你最欣赏的程序员有哪些？”在他的回答中，肯定会出现 John Carmack 的名字。没错，John Carmack 在程序员中的崇拜程度，甚至到了无以为加的地步，微软在很多问题上都非常尊重他的意见。正是因为对游戏和编程的热爱和孜孜不倦的追求，才使得高中毕业的 John Carmack 成为享誉世界的著名程序员，并开创了游戏领域的最大传奇。

第2部分

提高篇

第 9 堂课

Cookie 与会话控制

（视频讲解：145 分钟）

Cookie 和 SESSION 是数据的临时档案馆。Cookie 将数据存储在客户端，实现数据的持久存储。SESSION 将数据存储在服务器端，保证数据在程序的单次访问中持续有效。有了 Cookie 和 SESSION 这个临时档案馆，就可以解决 HTTP Web 协议的无状态问题，实现数据在不同页面之间的传递（如通过 SESSION 存储的数据来判断用户的访问权限）和数据在客户端的持久存储（如通过 Cookie 存储论坛用户的登录信息，实现用户下次在本机登录时，不需要输入用户名和密码，可以直接登录）。对于 Web 网站的开发，这是两项至关重要的内容，是读者必须掌握的知识。在本堂课中将对这两项内容进行系统的讲解。

学习摘要：

- **会话的操作**
- **会话的高级应用**
- **Cookie 的操作**

9.1　会话的操作

SESSION 译成中文为“会话”的意思，其本义是指有始有终的一系列动作/消息，如打电话时从拿起电话拨号到挂断电话这中间的一系列过程可以称之为一个 SESSION。

在计算机专业术语中，SESSION 是指一个终端用户与交互系统进行通信的时间间隔，通常指从注册进入系统到注销退出系统之间所经历的时间。SESSION 的工作原理如图 9.1 所示。

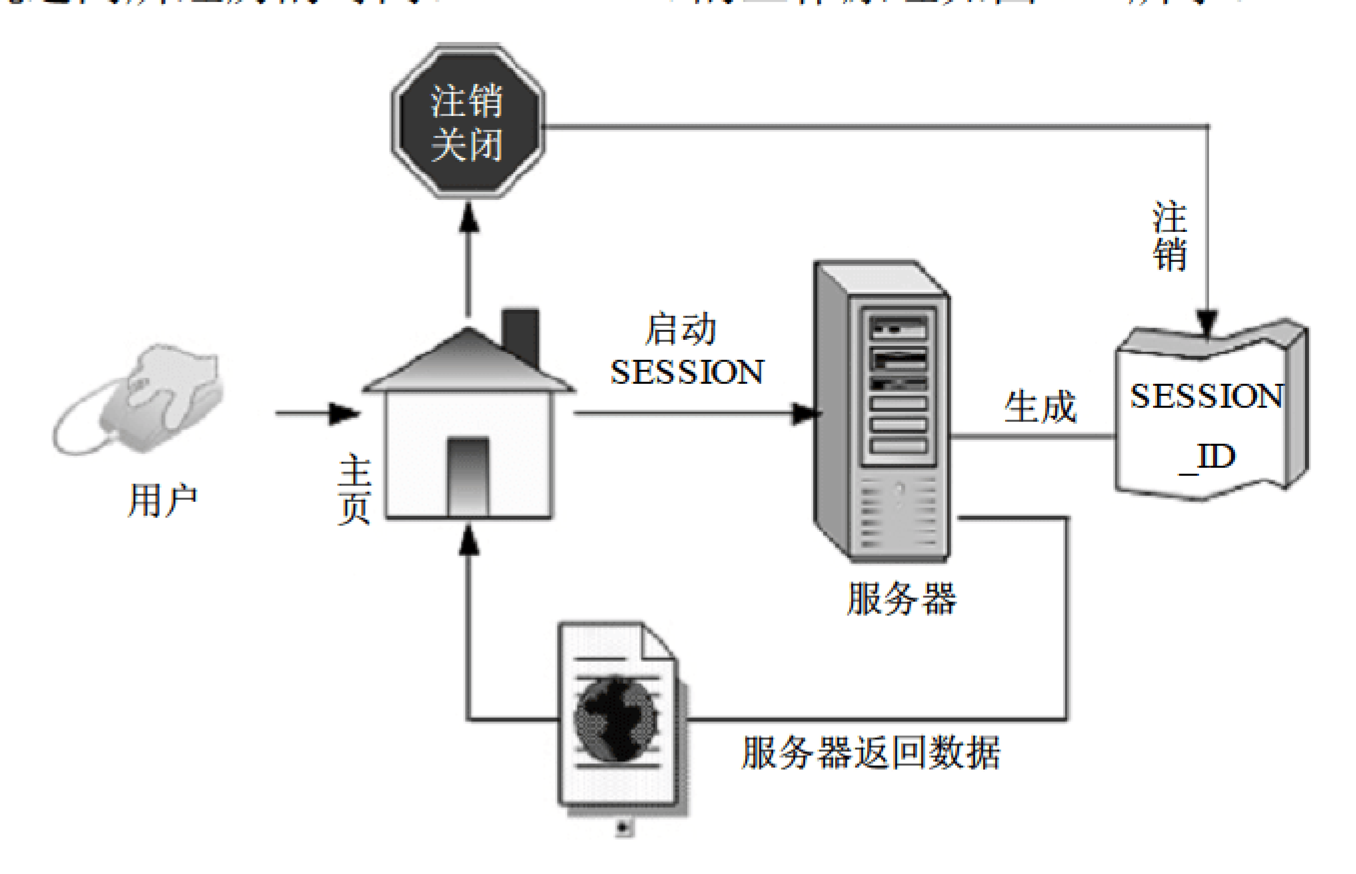


图 9.1　SESSION 工作原理

如图 9.1 所示，当登录网站时，启动 SESSION 会话，在服务器中随机生成一个唯一的 SESSION_ID，这个 SESSION_ID 在本次登录结束之前在页面中一直有效。当关闭页面或者执行注销操作后，该 SESSION_ID 会在服务器中自动注销。当重新登录此页面时，会再次生成一个随机且唯一的 SESSION_ID。

你问我答：SESSION 都具备哪些功能？

SESSION 在 Web 技术中占有非常重要的地位。由于网页是一种无状态的连接程序，所以无法获取用户的浏览状态。因此必须通过 SESSION 记录用户的有关信息，以供用户再次以此身份对 Web 服务器提供要求时做确认。例如，在电子商务网站中，通过 SESSION 记录用户登录的信息，判断用户的访问权限，通过 SESSION 实现购物车功能，记录用户所购买商品。

9.1.1　创建会话

通过以下 4 个步骤可以创建一个会话：启动会话、注册会话、使用会话和删除会话。下面对这 4 个步骤逐一进行介绍。

1．启动会话

启动 PHP 会话有两种方式：一种是使用 session_start()函数，另一种是使用 session_register()函数为会话登录一个变量来隐含地启动会话。

（1）通过 session_start()函数创建会话。该函数的语法如下：

```
bool   session_start(void)
```

该函数的参数为空。



注意：在使用 session_start()函数之前浏览器不能有任何输出，否则会产生类似于如图 9.2 所示的错误。即使在 PHP 脚本之外存在空格也不可以。

```
在session_start()函数之前输出的数据 Warning: session_start
(): Cannot send session cache limiter - headers already
sent (output started at F:\PkhPHP\www\MR\Instance\10
\10.1\index.php:2) in F:\PkhPHP\www\MR\Instance\10\10.1
\index.php on line 4
```

图 9.2 在使用 session_start()函数前输出字符串产生的错误

（2）通过 session_register()函数为会话登录一个变量来隐含地启动会话。该函数的语法如下：

```
boolean session_register(string name)
```

其中参数 name 用于指定新 session 变量的名称。使用 session_register()函数，要求将 php.ini 文件中的 register_globals 选项值设置为 On。

注意：如果应用 session_register()函数，则不必调用 session_start()函数，PHP 会在注册变量之后隐含地调用 session_start()函数。

注意：在 PHP 5.3.0 或者更高版本中不要使用 session_register()函数来启动会话。如果执意使用将出现如图 9.3 所示的错误信息。所以 PHP 会话的启动，推荐应用 session_start()。

```
Deprecated: Function session_register() is deprecated in
F:\PkhPHP\www\MR\Instance\10\10.1\index.php on line 2
Warning: session_register() expects at least 1
parameter, 0 given in F:\PkhPHP\www\MR\Instance\10\10.1
\index.php on line 2
```

图 9.3 错误地使用 session_ register()函数

2．注册会话

会话变量被启动后，全部保存在数组$_SESSION 中。通过数组$_SESSION 创建一个会话变量很容易，只要直接给该数组添加一个元素即可。

例如，启动会话，创建一个 SESSION 变量并赋予空值。代码如下：

```
<?php
session_start();                          //启动 SESSION
$_SESSION["user"] = null;                 //声明一个名为 user 的变量，并赋空值
?>
```

3．使用会话

例如，判断页面中是否存在会话 ID，如果不存在，就创建一个，并且使其能够通过全局数组$_SESSION 进行访问；如果已存在，则将这个已注册的会话变量载入以供用户使用。代码如下：

```
<?php
	session_start();
	if(isset(session_id())){
		echo $_SESSION['mr'];
	}else{
		$_SESSION['mr']="mrsoft";l
	}
?>
```

4. 删除会话

删除会话有 3 种方式，即删除单个会话、删除多个会话和结束当前会话。

☑ 删除单个会话

删除会话变量的方法同删除数组中的指定元素相同，都可以通过 unset()函数完成。unset()函数的语法如下：

```
unset ( $_SESSION['user'] ) ;
```

其参数是$_SESSION 数组中的指定元素，该参数不可以省略。通过 unset()函数一次删除数组中的一个元素；如果通过 unset()函数一次注销整个数组（unset($_SESSION)），那么会禁止整个会话功能，而且没有办法将其恢复，用户也不能再注册$_SESSION 变量。所以，如果读者要删除多个或全部会话，可以采用下面的两种方式。

☑ 删除多个会话

如果要一次注销所有的会话变量，可以将一个空的数组赋值给$_SESSION。代码如下：

```
$_SESSION = array() ;
```

☑ 结束当前会话

如果整个会话已经结束，那么就可以使用 session_destroy()函数结束当前的会话。session_destroy()函数的语法如下：

```
session_destroy();
```

该函数清空会话中的所有资源，彻底销毁 SESSION。

9.1.2 配置 PHP 的会话

在论坛中经常会看到对用户登录时间进行设置的复选框，该功能将您本次登录使用的用户名和密码保存在本机中，具体保存时间由用户自己选择，如保存一个星期、一个月等。

说明：用户在客户端实现上述功能，必须具备一个前提条件，即客户端必须支持 Cookie，否则一切都无从谈起。

1. 在客户端支持 Cookie 的前提下，控制 SESSION 的生命周期

在客户端支持 Cookie 的前提下，控制会话的生命周期有两种方法：一种是通过 SESSION 来完成，另一种是通过 Cookie 来完成。

下面介绍 SESSION 的生命周期，即 SESSION 的过期时间。SESSION 的过期时间设置有两种方法：第一种应用 session_set_Cookie_params()函数，第二种应用 setCookie()函数。

☑ session_set_Cookie_params()函数

该函数的功能是设置 SESSION 和 Cookie 的参数。其语法如下：

```
void session_set_Cookie_params( int lifetime [, string path [, string domain [, bool secure]]])
```

session_set_Cookie_params()函数的参数说明如表 9.1 所示。

表 9.1 session_set_Cookie_params()函数的参数说明

参　数	说　明
lifetime	必要参数。Cookie 的生存期
path	可选参数。Cookie 的有效路径
domain	可选参数。Cookie 的有效域
secure	可选参数。Cookie 在安全的范围内被发送

例如，在页面中创建一个 SESSION 变量，设置过期时间为 1 分钟。代码如下：

```
<?php
    $time=1*60;                                 //设定过期时间
    session_set_Cookie_params($time);           //设置 SESSION
    session_start();                            //启动 SESSION
    $_SESSION[id]="明日科技";                    //注册变量
```

注意：session_set_cookie_params()函数必须在 session_start()函数之前调用。由于该函数在不同的浏览器上使用会导致一些问题，所以不推荐使用这种方法来设置 SESSION 的过期时间。

☑ setCookie()函数

在 PHP 中，setCookie()函数的主要职责是创建 Cookie。但是，该函数同样可以控制 SESSION 的过期时间。有关 setCookie()函数的详细讲解请参考 9.3.1 节。这里直接通过一个实例，看如何利用 setCookie()函数设置 SESSION 的过期时间。演示代码如下：

```
<?php
    session_start();                            //启动 SESSION
    $time=1*60;                                 //设置 SESSION 过期时间
    setCookie(session_name(),session_id(),time()+$time,"/");        //设置 SESSION 过期时间
    $_SESSION[id]="明日科技";                                        //注册 SESSION
?>
```

说明：上述讲解的是在客户端支持 Cookie 的前提下如何设置 SESSION 的生命周期，下面介绍在客户端不支持 Cookie 的情况下，如何完成 SESSION 生命周期的设置。

2．在客户端不支持 Cookie 的前提下，控制 SESSION 的生命周期

一旦客户端禁止 Cookie，那么 SESSION 变量就不能够在页面间传递。这就好比在一家大型连锁超市办理了会员卡，但是超市之间并没有联网，那么会员卡就只能在办理的那家超市使用。解决该问题有 4 种方法：

☑ 在登录之前告知用户必须打开 Cookie，这是很多论坛的做法（暂且算一种方法）。

☑ 设置 php.ini 文件中的 session.use_trans_sid = 1 或者编译时打开-enable-trans-sid 选项，让 PHP 自动跨页面传递 SESSION_ID。

☑ 以 GET 或者 POST 方式，通过表单元素传递 SESSION_ID。

☑ 通过文件或者数据库传递 SESSION_ID。

上述 4 种方法，以第一种最为常用；第二种方法需要修改服务器中的 php.ini 文件，这种方法基本上不用，因为几乎不可能允许对服务器进行操作；第 3 种和第 4 种也是比较常用的方法，特别是第 4 种方法，修改 SESSION 数据的存储位置，可以减轻服务器的压力，特别适合于企业级网站开发使用。

9.2　会话的高级应用

9.2.1　SESSION 临时文件

在服务器中，如果将所有用户的 SESSION 都保存到临时目录中，会降低服务器的安全性和服务器的效率，导致打开服务器所在的站点会非常慢。解决该问题使用的就是 9.1.2 小节中的第 4 种方法，将 SESSION 存储到数据库或者临时文件中。

完成此项操作需要使用 PHP 中的 session_save_path()函数。session_save_path()函数用来取得或者重新配

置当前 SESSION 的存储路径。其语法如下：

```
string session_save_path([string path])
```

如果设置参数 path，表示重新设置 SESSION 的存储路径；如果不设置参数 path，表示直接获取当前 SESSION 的存储路径。

例如，将 session 文件保存在当前目录的 session_files 下，代码如下：

```
<?php
$path = './session_files/';                    //设置 SESSION 存储路径
session_save_path($path);                      //重新设置 SESSION 的存储路径
session_start();                               //初始化 SESSION
$_SESSION[id] = "明日科技";
?>
```

9.2.2 SESSION 缓存

所谓 SESSION 的缓存是指将网页中的内容临时存储到客户端 IE 的 Temporary Internet Files 文件夹下，并且可以设置缓存的时间。当网页第一次被浏览后，页面的部分内容在规定的时间内被临时存储在客户端的临时文件夹中，这样在下次访问该页面时，就可以直接读取缓存中的内容，从而提高网站的浏览效率。

1．缓存函数——session_cache_limiter

实现 SESSION 的缓存功能应用的是 session_cache_limiter()函数，其语法如下：

```
string session_cache_limiter ( [string cache_limiter])
```

参数 cache_limiter 用于设置缓存的方式，参数值的说明如表 9.2 所示。

表 9.2　cache_limiter 参数值的说明

参　数　值	说　　明
nochache	不设置缓存
private	私有方式
private nochache	私有方式，但不过期
public	公有方式

2．缓存时间函数——session_cache_expire

设置缓存时间应用的是 session_cache_expire()函数，其语法如下：

```
int session_cache_expire ( [int new_cache_expire])
```

参数 cache_expire 为可选参数，设置 SESSION 的过期时间，单位为分钟。默认过期时间是 180 分钟（即不设置参数值的情况下）。

提示： session_cache_limiter()和 session_cache_expire()函数必须在 session_start()函数调用之前使用，否则将出错。

例如，应用 session_cache_limiter()函数设置缓存方式，应用 session_cache_expiry()函数设置缓存时间，最后，分别输出缓存方式和缓存时间。代码如下：

```
<?php
    session_cache_limiter("private");          //设置缓存方式
    session_cache_expire(1);                   //设置缓存时间
    session_start();                           //启动会话
```

```
        echo "缓存方式：".session_cache_limiter();      //输出缓存方式
        echo "<br>";
        echo "缓存时间:".session_cache_expire();        //输出缓存时间
?>
```

运行结果如图 9.4 所示。

查看客户端缓存文件夹（Temporary Internet Files），可以看到如图 9.5 所示的内容。

SESSION缓存

缓存方式：private
缓存时间:1

图 9.4　SESSION 客户端缓存



图 9.5　查看到的缓存文件

注意：虽然通过缓存时间函数设置缓存过期时间，但是在过期之后并不能够删除存储在客户端缓存文件夹中的缓存文件，缓存文件的删除由 Windows 操作系统自行控制，或者由用户手动删除。

9.2.3　SESSION 数据库存储

将 SESSION 存储于指定文件中，虽然不至于将临时文件夹填满而造成站点瘫痪，但是如果一个大型网站一天登录 1200 人，一个月登录 30000 人，这时站点中存在 30000 个 SESSION 文件，要在这 30000 个文件中查询一个 SESSION_ID，速度可想而知。如果此时将 SESSION_ID 存储于数据库中，那么查询的速度受到的影响就会小很多。

将 SESSION_ID 存储于数据库中，应用的是 session_set_save_handler()函数。它调用自定义函数，完成将 SESSION_ID 存储于数据库中的操作。该函数的语法如下：

```
bool session_set_save_handler ( string open, string close, string read, string write, string destroy, string gc)
```

session_set_save_handler()函数的参数说明如表 9.3 所示。

表 9.3　session_set_save_handler()函数的参数说明

参　　数	说　　明
open(save_path,session_name)	找到 SESSION 存储地址，取出变量名称
close()	不需要参数，关闭数据库
read(key)	读取 SESSION 键值，key 对应 SESSION_ID
write(key,data)	其中 data 对应设置的 SESSION 变量
destroy(key)	注销 SESSION 对应的 SESSION 键值
gc(expiry_time)	清除过期 SESSION 记录

这是一个非常特殊的函数，因为一般函数的参数都是变量，但是此函数的参数为 6 个函数。通过这 6 个函数完成 SESSION 的数据库存储。

9.3　Cookie 的操作

Cookie 是一种在远程客户端存储数据并以此来跟踪和识别用户的机制。简单地说，Cookie 是 Web 服务

器暂时存储在用户硬盘上的一个文本文件，并随后被 Web 浏览器读取。当用户再次访问 Web 网站时，网站通过读取 Cookie 文件记录这位访客的特定信息（如上次访问的位置、花费的时间、用户名和密码等），从而迅速作出响应。例如，在页面中不需要输入用户的 ID 和密码，即可直接登录网站。

这个文本文件以此种格式（用户名@网站地址[数字].txt）存储于客户端硬盘的指定文件夹下。例如，客户端机器为 Windows 2000/XP/2003 操作系统，系统盘为 C 盘，当通过 IE 浏览器访问 Web 网站时，Web 服务器会自动以用户名@网站地址[数字].txt 格式生成 Cookie 文本文件，并存储在用户硬盘的指定位置，如图 9.6 所示。



图 9.6　Cookie 文件的存储路径

> **注意**：在 Cookies 文件夹下，每个 Cookie 文件都是普通的文本文件，而不是程序。文本文件中的内容大多都经过了加密处理，因此，表面看来只是一些字母和数字的组合，而只有服务器的 CGI 处理程序才知道它们真正的含义。

在本节的开始介绍了什么是 Cookie，Cookie 都可以实现哪些功能，下面开始进入正题，讲解如何创建、访问和删除 Cookie。

9.3.1　创建 Cookie

创建 Cookie 应用的是 setCookie()函数。由于 Cookie 是 HTTP 头标的组成部分，作为头标必须在页面其他内容之前发送，也必须最先输出，所以在 setCookie()函数之前不能有任何内容输出，即使是一个 HTML 标记、一个 echo 语句甚至一个空行都会导致程序出错，这就是 setCookie()函数，语法如下：

```
bool setCookie(string name[,string value[,int expire[, string path[,string domain[,int secure]]]]])
```

setCookie()函数的参数说明如表 9.4 所示。

表 9.4　setCookie()函数的参数说明

参　数	说　明	举　例
name	Cookie 的变量名	可以通过$_COOKIE['Cookiename ']调用变量名为 Cookiename 的 Cookie
value	Cookie 变量的值，该值保存在客户端，不能用来保存敏感数据	可以通过$_COOKIE['values ']获取名为 values 的值
expire	Cookie 的过期时间，expire 是标准的 UNIX 时间标记，可以用 time()函数或 mktime()函数获取，单位为秒	如果不设置 Cookie 的过期时间，那么 Cookie 将永远有效，除非手动将其删除
path	Cookie 在服务器端的有效路径	如果该参数设置为“/”，则它在整个 domain 内有效；如果设置为“/12.9”，则它在 domain 下的/12.9 目录及子目录内有效。默认是当前目录

续表

参　数	说　明	举　例
domain	Cookie 有效的域名	如果要使 Cookie 在 mrbccd.cn 域名下的所有子域都有效，应该设置为 mrbccd.cn
secure	指明 Cookie 是否仅通过安全的 HTTPS，值为 0 或 1	如果值为 1，则 Cookie 只能在 HTTPS 连接上有效；如果值为默认值 0，则 Cookie 在 HTTP 和 HTTPS 连接上均有效

在了解了 Cookie 的创建方法后，下面编写一段代码并利用 setCookie()函数来创建一个 Cookie。代码如下：

```
<?php
setCookie("mr",'明日科技');
setCookie("mr", '明日科技', time()+60);                    //设置 Cookie 有效时间为 60 秒
//设置有效时间为 60 秒，有效目录为“/12.8/”，有效域名为 mrbccd.cn 及其所有子域名
setCookie("mr", "明日科技", time()+3600, "/12.8/",". mrbccd.cn", 1);
echo "读取 Cookie：".$_COOKIE['mr'];                       //通过$_COOKIE[]读取 Cookie 的值
?>
```

在客户端的 Cookies 文件夹下会自动生成一个 Cookie 文件，名为 administrator@xxx.txt，Cookie 的有效期为 60 秒，失效后 Cookie 文件自动删除。

9.3.2　访问 Cookie

在 PHP 中应用全局数组$_COOKIE[]读取客户端 Cookie 的值。$_COOKIE 是经由 HTTP Cookies 方法提交至脚本的变量。类似于旧数组$HTTP_Cookie_VARS（依然有效，但不推荐使用）。通过$_COOKIE 获取 Cookie 值已经在前面使用过，这里不再重复。

说明： 在使用 Cookie 时要注意，如果未设置 Cookie 的过期时间，那么在关闭浏览器时会自动删除 Cookie 数据。如果设置 Cookie 的过期时间，那么浏览器将会保存 Cookie 数据，即使用户重新启动计算机，只要没有过期，Cookie 数据就一直有效。

9.3.3　删除 Cookie

前面已经了解如何创建和访问 Cookie，如果 Cookie 被创建后，没有设置过期时间，那么 Cookie 文件会在浏览器关闭时自动删除。但是，要在关闭浏览器之前删除 Cookie 文件，应该怎么办呢？方法有两种：一种是使用 setCookie()函数删除，另一种是在客户端手动删除。

1．使用 setCookie()函数删除 Cookie

删除 Cookie 只需要将 setCookie()函数中的第二个参数设置为空值，将第 3 个参数（Cookie 的过期时间）设置为小于系统的当前时间即可。

例如，可以将 Cookie 的过期时间设置为当前时间减 1 秒，就实现了删除 Cookie 的操作。代码如下：

```
setCookie("mr", "", time()-1);
```

提示： 还可以直接将过期时间设置为 0，即直接删除 Cookie。

2．在客户端手动删除 Cookie

大家知道，Cookie 被创建后，会自动生成一个文本文件存储在客户端 IE 浏览器的 Cookies 临时文件夹中。因此就可以通过操作客户端 IE 浏览器的 Cookies 临时文件夹删除 Cookie。具体操作步骤如下：

选择 IE 浏览器中的“工具”/“Internet 选项”命令，打开“Internet 选项”对话框，如图 9.7 所示。在“常规”选项卡中单击“删除 Cookies”按钮，将弹出如图 9.8 所示的“删除 Cookies”对话框，单击“确定”按钮，即可成功删除全部 Cookie 文件。



图 9.7 “Internet 选项”对话框



图 9.8 “删除 Cookies”对话框

9.4 照猫画虎——基本功训练

9.4.1 基本功训练 1——初始化 SESSION 变量

视频讲解：光盘\mr\09\lx\初始化 SESSION 变量.exe

实例位置：光盘\mr\09\zmhh\01

初始化 SESSION 的 SESSION_ID 大家可以把它看成是某健身会馆的 VIP 会员卡，在此健身会馆内只要有会员卡，用户即可根据自己的需要体验任何一种健身器材。SESSION 也是一样，只要启动了 SESSION，那么 SESSION 变量在任何一个页面都可以引用。用户初始化 SESSION 非常简单，其实现过程如下：

（1）通过 session_start()函数启动 SESSION 会话。

（2）通过$_SESSION[]全局数组注册一个 SESSION 变量。实例代码如下：

```
<?php
    session_start();                    //启动 SESSION
    $_SESSION['mr']='mrsoft';           //初始化 SESSION 变量
?>
```

照猫画虎：在上面的页面中，再添加一个 SESSION 变量，让它的变量名为“明日科技”，变量值为 mrkj。（20 分）（实例位置：光盘\mr\09\zmhh\01_zmhh）

9.4.2 基本功训练 2——通过 SESSION 变量在页面间传递数据

视频讲解：光盘\mr\09\lx\通过 SESSION 变量在页面间传递数据.exe

实例位置：光盘\mr\09\zmhh\02

利用 SESSION 在页面间传递数据是 SESSION 的主要应用，例如，当用户注册登录后，主页面显示注



册用户的用户名并标有温馨提示语句。这就是 SESSION 变量在页面间传递数据的一个典型应用。下面将实现这一过程。运行结果如图 9.9 和图 9.10 所示。

用户名：
密　码：
登录　重置

图 9.9　表单登录

欢迎XXXX光临，祝您愉快！

图 9.10　输出提示信息

具体步骤如下：

（1）创建两个脚本文件，分别命名为 index.php 和 in.php。

（2）在 index.php 文件中启用 SESSION 并编写一个表单登录框，将登录框的用户名保存在 SESSION 中。

（3）在 in.php 页面中接收 SESSION 变量并添加提示信息。代码如下：

```
<!--index.php 页-->
<?php
	session_start();								//启动 SESSION
	if(isset($_POST[sub])){							//判断页面中是否存在 post 变量
		if($_POST[text]!="" and $_POST[pwd]!=""){			//判断文本框是否为空
			$_SESSION['user']=$_POST[text];				//注册会话
			echo "<script>window.location.href='in.php';</script>";	//跳转
		}else{
			echo "<script>alert('文本内容不能为空')</script>";		//提示
		}
	}
?>
<!--in.php 页-->
<?php
			if(isset($_SESSION['user'])){					//如果页面中存在 SESSION 变量
				echo "欢迎<b>".$_SESSION['user']."</b>光临，祝您愉快!";	//输出提示信息
			}else{
				echo "<script>alert('页面不存在 SESSION 变量');</script>";	//提示信息
			}
?>
```

照猫画虎：在上述实例中，将用户密码也保存在 SESSION 变量中，然后在 in.php 页面中输出提示信息。（20 分）（实例位置：光盘\mr\09\zmhh\02_zmhh）

9.4.3　基本功训练 3——通过 SESSION 变量判断页面的访问权限

视频讲解：光盘\mr\09\lx\通过 SESSION 变量判断页面的访问权限.exe

实例位置：光盘\mr\09\zmhh\03

利用 SESSION 变量判断页面的访问权限与页面间传递数据大同小异，只是在原有基础上判断页面中的 SESSION 变量是否具备访问权限。运行结果如图 9.11 和图 9.12 所示。

具体步骤如下：

（1）创建两个脚本文件，分别命名为 index.php 和 in.php。

（2）在 index.php 文件中启用 SESSION 并编写一个表单登录框，将登录的用户名和密码保存在

SESSION 变量中。

图 9.11　管理员登录信息　　　　图 9.12　管理员登录提示

（3）在 in.php 页面中接收 SESSION 变量传递的数据，并判断与指定的 SESSION 变量是否相同，如果相同输出对管理员的温馨提示，否则输出对普通会员的温馨提示。代码如下：

```
<!--index.php 页-->
<?php
    session_start();                                                    //启动 SESSION
    if(isset($_POST[sub])){                                             //判断页面中是否存在 post 变量
        if($_POST[text]!="" and $_POST[pwd]!=""){                       //判断文本框是否为空
            $_SESSION['user']=$_POST[text];                             //注册会话
            $_SESSION['pwd']=$_POST[pwd];
            echo "<script>window.location.href='in.php';</script>";    //跳转
        }else{
            echo "<script>alert('文本内容不能为空')</script>";           //提示
        }
    }
?>
<!--in.php 页-->
<?php
            if(isset($_SESSION['user'])){                               //如果页面中存在 SESSION 变量
                if($_SESSION['user']=="mr" and $_SESSION['pwd']=="mrsoft"){
                                                                        //判断是否与指定的 SESSION 变量相同
                    echo "欢迎管理员<b>".$_SESSION['user']."</b>的归来";          //输出管理员提示
                }else{
                    echo "欢迎<b>".$_SESSION['user']."</b>光临，祝您愉快!";       //输出普通用户提示
                }
            }else{
                echo "<script>alert('页面不存在 SESSION 变量');</script>";      //输出提示
            }
    ?>
```

照猫画虎：依据上述实例，添加超级管理员的权限，并在 in.php 页面中输出对超级管理员的温馨提示信息。（20 分）（实例位置：光盘\mr\09\zmhh\03_zmhh）

9.4.4　基本功训练 4——Cookie 控制网页访问时间

视频讲解：光盘\mr\09\lx\Cookie 控制网页访问时间.exe

实例位置：光盘\mr\09\zmhh\04

个人的 PC 电脑已经进入千家万户，这在大多数家长的眼里也许并不是一件好事，因为如果孩子沉迷于网络会影响到学习。所以控制孩子使用电脑的时间就变得尤为重要，控制电脑的运行时间需要系统级软件来完成，是 PHP 很难做到的，但是控制页面的访问时间对于 PHP 来说简直是轻而易举。下面将实现这一功能。运行结果如图 9.13 和图 9.14 所示。



Cookie生命周期并未完结 距Cookie失效还有8秒

图 9.13　Cookie 的过期时间

Cookie生命周期结束

图 9.14　Cookie 结束生命周期

具体步骤如下：

（1）创建一个脚本文件，命名为 index.php。

（2）利用 setCookie()函数定义一个 Cookie 变量，并为此变量设置失效时间。当超过指定时间后页面显示提示信息，告知用户页面超时。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");				//设置编码格式
	$time=1*10;									//设置过期时间
	setCookie("mr","mrsoft",time()+$time);						//设置一个 Cookie 变量
	setCookie("time",$time,time()+$time);						//设置一个 Cookie 变量
		if(isset($_COOKIE['mr'])){							//如果存在 Cookie 变量 mr
		echo "Cookie 生命周期并未完结";						//输出提示信息
	}else{
		echo "Cookie 生命周期结束";
	}
?>
```

照猫画虎：在上述实例基础上，将 Cookie 生命周期的控制全交给用户，实现用户自定义控制网页的过期时间。（20 分）（实例位置：光盘\mr\09\zmhh\04_zmhh）

9.4.5　基本功训练 5——通过 Cookie 保存用户登录信息

视频讲解：光盘\mr\09\lx\通过 Cookie 保存用户登录信息.exe

实例位置：光盘\mr\09\zmhh\05

开心网相信大家都很熟悉，它的主页登录界面就采用了 Cookie 保存用户登录信息的机制。将用户的用户名和密码根据用户指定的时间长期保存在客户端的 PC 上。下面将实现利用 Cookie 保存用户登录信息的功能。运行结果如图 9.15 所示。

用户名：mrsoft
密　码：●●●●●●
1小时　1天　1周
登录　重置

图 9.15　Cookie 保存用户登录信息

具体步骤如下：

（1）创建两个脚本文件，分别命名为 index.php 和 in.php。

（2）利用 setCookie()函数定义两个变量，将用户名和密码的值赋给这两个变量并根据用户操作指定失效时间。当超过失效时间后用户的登录信息将不再保存；否则，程序会自动记录用户的登录信息。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");					//定义编码格式
	setCookie('text',$_POST[text],time()+$_POST[radio]);				//设置 Cookie 变量
	setCookie('pwd',$_POST[pwd],time()+$_POST[radio]);
?>
<form action='in.php' method='post'>							//将 Cookie 的值赋给文本框
		用户名：<input class="two" type="text" name="text" value="<?php echo $_COOKIE[text];?>"><br>
	密  码：<input class="two"type="password" name="pwd" value="<?php echo $_COOKIE
[pwd]?>"><br>
```

```
</form>
<?php
	if(isset($_POST[sub])){                                         //当页面中存在变量 sub 时
		if($_POST[text]=="" and $_POST[pwd]==""){                    //判断并输出提示
			echo "<script>alert('文本框内容为空');</script>";
		}
	}
?>
```

照猫画虎：在上述实例基础上，去掉 Cookie 自动保存用户登录信息的功能。（20 分）（实例位置：光盘\mr\09\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

9.5　情景应用——拓展与实践

9.5.1　情景应用 1——对 SESSION 变量编码和解码

视频讲解：光盘\mr\09\lx\对 SESSION 变量编码和解码.exe

实例位置：光盘\mr\09\qjyy\01

虽然 SESSION 是一种服务器端的机制，但是任何一种有利于系统安全的程序是程序员都不能放弃的。笔者绝不否认 SESSION 的安全性，但是出于各种需要，用户可能希望将 SESSION 变量保存于客户端或者数据库中。当需要这些操作时用户可以将 SESSION 进行编码，然后根据需要再将其解码。为模仿这一过程，笔者将 SESSION 的相关变量保存在客户端的文本文件中，并对其进行编码和解码操作。运行结果如图 9.16 和图 9.17 所示。

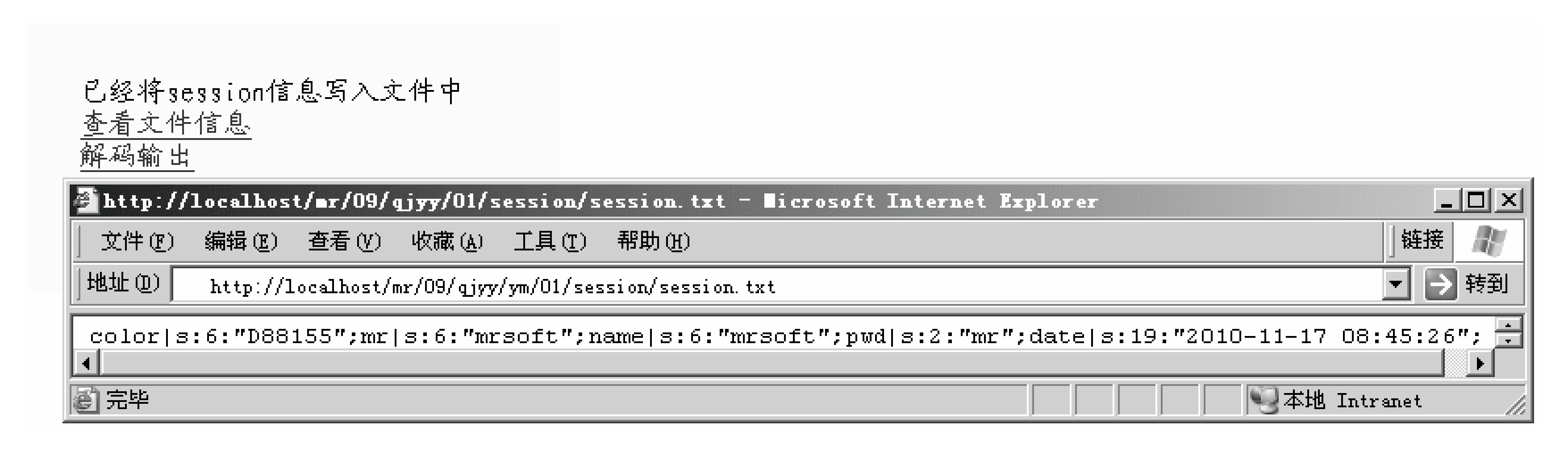


图 9.16　将 SESSION 编码并读取文件信息

已经将session信息写入文件中
查看文件信息
解码输出
用户名：mrsoft
密码：mr
日期\时间：2010-11-17 01:32:08

图 9.17　将文件信息中的 SESSION 变量解码并输出



具体步骤如下：

（1）创建脚本文件 index.php。

（2）通过 mkdir()函数在客户端创建 session.txt 文本文件，并对 SESSION 变量进行编码，然后保存到文本文件中。最后，对数据进行解码并输出。代码如下：

```
<?php
    session_start();                                            //启用 SESSION
    $_SESSION['name']='mrsoft';                                 //注册会话
    $_SESSION['pwd']='mr';
    $_SESSION['date']=date('Y-m-d H:i:s');
    if(!is_dir("session")){                                     //判断文件夹是否存在
        mkdir("session");                                       //新建文件夹
    }
    $string=session_encode();                                   //将 SESSION 进行编码
    if(file_put_contents("session/session.txt",$string)>0){     //写入文件
        echo "已经将 session 信息写入文件中<br>";
        echo "<a href='session/session.txt'>查看文件信息</a><br>";
    }
    $str=file_get_contents("session/session.txt");              //读出文件信息
    session_decode($str);                                       //SESSION 解码
    echo "<a href='index.php?id=1'>解码输出</a>";
    if($_GET[id]==1){
        echo "<br>用户名：".$_SESSION['name']."<br>";             //输出 SESSION 信息
        echo "密码：".$_SESSION['pwd']."<br>";
        echo "日期\时间：".$_SESSION['date']."<br>";
    }
?>
```

DIY：将 SESSION 变量编码后存入数据库，然后解码输出到页面。（20 分）（实例位置：光盘\mr\09\qjyy\01_diy）

9.5.2　情景应用 2——通过 SESSION 变量屏蔽刷新页面对计数器的影响

视频讲解：光盘\mr\09\lx\通过 SESSION 变量屏蔽刷新页面对计数器的影响.exe

实例位置：光盘\mr\09\qjyy\02

计数器是用来保存用户访问网站统计数量的，使用计数器的好处在于网站的管理者可以很直观地看到当日或当月的访问总数，实现探查可能的潜在用户和规划网站的发展战略等。另一方面，高访问量的网站计数器本身也是一种免费的广告效应，吸引潜在用户加入其中。本实例的运行结果如图 9.18 所示。

网站计数器
本网站已被访问3次

图 9.18　网站计数器

具体步骤如下：

（1）创建脚本文件，命名为 index.php。

（2）启用 SESSION，利用 microtime()函数定义当前时刻时间戳的微秒数，并将此微秒数分别保存在 SESSION 和隐藏域文本框中。

（3）当用户每刷新一次页面或开启浏览器进入到页面时，首先判断隐藏域的值是否与 SESSION 变量的值相同，如果相同则程序向下执行，并将获取到的值加 1 后保存于文本文件中；否则页面为用户自定义刷新，文本文件的值不加 1 输出。代码如下：

```
        <form action=""method="post">                  //定义隐藏域
            <input type="hidden" name="hidden" value="<?php echo $time;?>">
        </form>
<?php
        if($_SESSION[time]==$_POST[time]){             //检测页面中$_SESSION[time]是否与$_POST[time]相同
            $visit=file_get_contents("index");         //取得文本文件信息
            $visit=$visit+1;                           //将值自动加 1
            echo "<font>本网站已被访问<b>".$visit."</b>次</font>";   //输出提示信息
            file_put_contents("index",$visit);                      //将加 1 后的值重新写入文件
        }else{
            $visit=file_get_contents("index");                      //获取到文本文件的值
            echo "<font>本网站已被访问<b>".$visit."</b>次</font>";   //输出信息
        }
        $time=microtime();                                          //获取当前时刻时间戳的微秒数
        $_SESSION[time]=$time;                                      //将此微秒数保存在 SESSION 中
?>
```

DIY：本实例是将 SESSION 信息保存在文本文件中，用户可以根据需求将其保存在数据库中并实现相同功能。（20 分）（实例位置：光盘\mr\09\qjyy\02_diy）

9.5.3 情景应用 3——通过 Cookie 计算在线考试答题时间

视频讲解：光盘\mr\09\lx\通过 Cookie 计算在线考试答题时间.exe

实例位置：光盘\mr\09\qjyy\03

Cookie 控制在线考试答题时间在实际的应用开发中并不十分常见，当然 Cookie 实现这一功能是绰绰有余的。实现本实例的核心原理是：当用户进入到页面时利用 setCookie()函数保存当前时间戳，然后当用户答题完毕触发按钮后获取此时刻的时间戳，并与 Cookie 变量存储的时间戳做差运算，计算出考试用时。运行结果如图 9.19 所示。

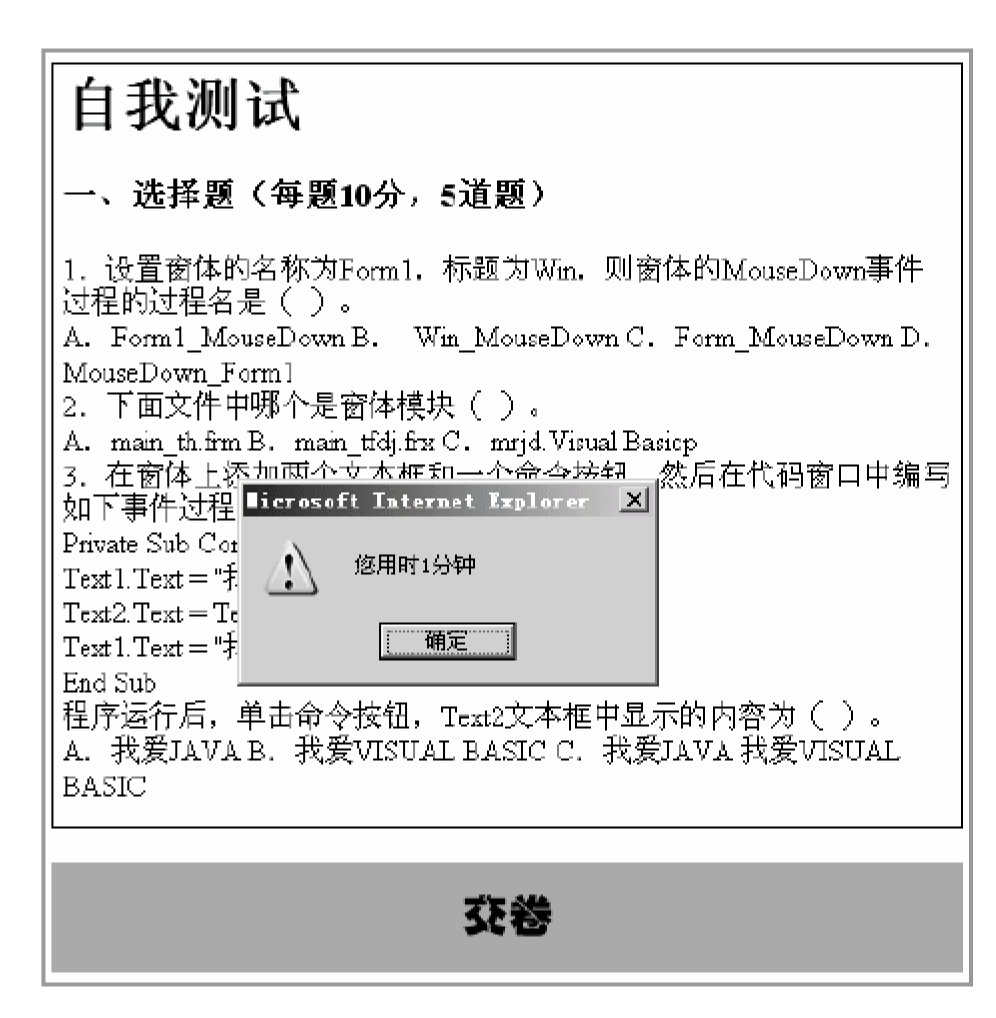


图 9.19 控制在线考试答题时间

具体步骤如下：

（1）创建脚本文件，命名为 index.php。



（2）利用 file_get_contents()函数获取文本文件 index.txt 的数据信息，与此同时获取此时刻的时间戳并保存在 Cookie 之中。

（3）编写表单，定义一个提交按钮，当按钮被单击后获取完成答题时刻的时间戳，然后与 Cookie 变量存储的时间戳做差运算，取得答题所用时间。代码如下：

```
<?php
	$time_bgn=time();                          //获取当前时间戳
	setCookie("time_bgn",$time_bgn);           //将时间戳保存在 Cookie 中
	echo file_get_contents('index');           //输出文本文件信息
?>
<form action="" method="post">                 //定义表单信息
	<input type="submit" name="button" value="交卷">
</form>
<?php
	if($_POST[button]){                        //当提交按钮被单击后
		$time=time();                          //获取此时刻的时间戳
		$time_fini=($time-$_COOKIE['time_bgn']-10);  //取得考试所用时间
?>
//输出数据
		<input class="one" type='hidden' value='<?php echo "您用时".ceil($time_fini/60)."分钟";?>'>
		<script>alert($(".one").attr('value'));</script>
<?
	}
?>
```

DIY：本实例是将当前时间戳保存在 Cookie 中计算考试时间，用户可以根据需求获取当前时间戳的微秒数，计算考试时间以达到精确的目的。（20 分）（实例位置：光盘\mr\09\qjyy\03_diy）

9.5.4　情景应用 4——Cookie 自动登录

视频讲解：光盘\mr\09\lx\Cookie 自动登录.exe

实例位置：光盘\mr\09\qjyy\04

自动登录在网站的开发中也是非常常见的，例如，PHP100 中文网站就采取了此种机制，实现 Cookie 生命周期范围内的自动登录。其实实现这一功能非常简单，就是判断当前页面中是否存在一个 Cookie 变量，如果存在则自动登录，否则重新登录。本实例就实现了这个功能，运行结果如图 9.20 和图 9.21 所示。



图 9.20　填写登录信息

图 9.21　自动登录

具体步骤如下：

（1）创建脚本文件 index.php，首先判断页面中是否存在 Cookie 变量 user，如果存在则实现自动登录。

（2）当页面中不存在 Cookie 变量 user 时显示表单内容，提示用户需登录。核心代码如下：

```
    <?php
setCookie("user",$_POST[text],time()+3600);                    //注册 Cookie 变量
        if($_COOKIE[user]!=""){                                //如果页面中 Cookie 不为空
            echo "<div class='e'><font>欢迎".$_COOKIE[user]."光临！ </font></div>";
        }else{
?>
        //表单
        <form action="" method="post">
            用户名：<input class="one" type="text" name="text">
            密码：<input class="one"type="password" name="pwd">  
            <input  type="submit"class="two"  name="sub"  value="登录">  <input  class="two"type
="reset" name="res" value="重置">
        </form>
    <?php
            if(isset($_POST[sub])){                            //当提交按钮被单击时
                if($_POST[text]!="" and $_POST[pwd]!=""){      //判断文本框是否为空
                    echo "<script>alert('登录成功');</script>";
                }
            }
        }
    ?>
```

DIY：本实例是判断页面中是否存在一个 Cookie 用户名变量，用户可以根据需求添加一个 Cookie 密码变量并在欢迎语中将密码显示为“*”。（20 分）（光盘\mr\09\qjyy\04_diy）

9.5.5　情景应用 5——通过 SESSION 控制页面的背景色

视频讲解：光盘\mr\09\lx\通过 SESSION 控制页面的背景色.exe

实例位置：光盘\mr\09\qjyy\05

网页换肤在网站开发中只是一个微不足道的“小玩意”，但是对于初学者来说，它会给用户带来巨大的成就感。下面将实现这一功能，运行结果如图 9.22 所示。



图 9.22　SESSION 控制背景色

具体步骤如下：

（1）创建脚本文件 index.php，首先判断页面中是否存在 SESSION 变量 color，如果存在则网页不变色。

（2）当页面中不存在 SESSION 变量 color 时重新注册 SESSION 变量并实现动态变色。核心代码如下：

```
<?php
    $time=3600;                                            //定义时间
    setCookie(session_id(),session_name().time()+$time);   //定义 SESSION 失效时间
    session_start();                                       //启动 SESSION
    if(!isset($_SESSION['color'])){                        //如果页面中存在 SESSION 变量 color
        $array=array();                                    //定义空数组
        for($a=0;$a<6;$a++){                               //循环语句
            $rand=rand(1,16);                              //获取 1～16 的随机数
            if($rand>10){                                  //如果随机数大于 10
```

```
			$rand=strtoupper(dechex($rand));		//转换为大写的十六进制数
		}
		if($rand==10){				//如果随机数等于 10，则此随机数等于 A
			$rand="A";
		}
		array_push($array,$rand);			//将每一个随机数压入数组
	}
	$string=implode($array);				//合成字符串
	$_SESSION['color']=$string;			//将字符串保存在 SESSION 中
?>
```

DIY：本实例是判断页面中是否存在一个 SESSION 变量，用户可以根据此方法动态改变标题“SESSION 控制背景颜色”的字体颜色。（20 分）（光盘\mr\09\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

9.6　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．SESSION 是（　　）。

A．客户端机制　　B．服务器端机制　　C．网页机制

2．Cookie 是（　　）。

A．客户端机制　　B．服务器端机制　　C．网页机制　　D．MouseDown_Form1

3．下面文件中（　　）是启动 SESSION 的语句。

A．session_start()　　B．SESSION_START()　　C．setCookie()

4．判断某个 SESSION 或 Cookie 变量使用（　　）函数。

A．is_dir()　　B．is_int()　　C．isset()

5．看下面代码：

```
	session_start();
$_SESSION['mr']='明日科技';
```

输出$_SESSION['mr']变量的值为（　　）。

A．mr　　B．MR　　C．明日科技

二、填空题（每题 10 分，5 道题）

1．判断页面是否存在'mr'的 session 变量，请将代码补充完整。

```
	session_start();
	$_SESSION['mr']='明日科技';
	If（　　　　　　　　　　　　）
```

2．定义一个名称为'mr'值为'mrsoft'的 Cookie 变量，应该写为（　　　　）。

3．获取 2 题中 Cookie 变量'mr'，应该写为（　　　　）。

4．为了删除所有 SESSION 变量，应把它设置为（　　）。

5．程序运行后，页面显示的内容为（　　）。

```
setCookie("mr","mrsoft",time()+3600);
echo $_COOKIE['mr'];
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

9.7　行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）利用 SESSION 做一个防刷新注册页面。	
		（2）编写一个主页文件并实现网页换肤的操作。	
		（3）将 SESSION 的相关数据保存在数据库中实现对 SESSION 数据的控制。	
		（4）利用 SESSION 和数据库知识实现仿 Cookie 自动登录。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

9.8　成功可以复制——微型博客创始人埃文·威廉姆斯

成就：Twitter 首席执行官，创建于 2006 年的 Twitter，目前用户数量超过 1.65 亿。创始博客网（Blogger），后卖给 Google。

创始公司：Twitter 公司，全球第三大社交网络平台

财富巅峰：拥有市值 10 亿美元的 Twitter 公司绝大部分股份

最高学历：高中

出生日期：1970 年 8 月 20 日

超级语录：企业家分为两种类型，一种是把赚钱当作目标，另一种是为了创造新事物，创造从未有过的产品或服务。我属于第二种，并热衷于此。这是世界上出现新事物的源泉，而我非常幸运能够创造出一个改变世界的产品。

成功历程

Twitter，这个发布于 2006 年的微型博客网站，目前已成为互联网时代最新最热的明星。对很多人来说，Twitter 的迅速崛起实在令人意外，对 Twitter 的创始人埃文·威廉姆斯来说，同样颇为感慨。

要知道，他当初推出 Twitter，主要是为了方便交流和共享。

埃文·威廉姆斯，1970 年 8 月 20 日出生在内布拉斯加州的一个农场，高中毕业后，考入了内布拉斯加大学，中途认为“上大学是浪费时间”而退学。1994 年，埃文·威廉姆斯创办了一家互联网公司。虽然埃文·威廉姆斯当时并不懂互联网，但他认为互联网最终将非常重要。通过自学，埃文·威廉姆斯知道了如何创建网站，并给一些企业做一些项目。但因没有管理经验，企业没经营多久就倒闭了。回忆过去，埃文·威廉姆斯说：“当时我赔了很多钱，包括我父亲的投资。公司因为拖欠国税局的税款而关门，当时我的很多员工都要疯掉了。”



在第一次创业失败后，埃文·威廉姆斯并没有放弃，他决定去硅谷闯荡。威廉姆斯进入一家媒体公司，一开始做销售，后来开始编写电脑程序，很快，他就开始承接硅谷巨头英特尔公司和惠普公司的业务。但他厌倦了公司的环境，决定辞职继续创业。

1999 年，埃文·威廉姆斯与人合伙成立了 Pyra 实验室，主要制作管理软件。在这期间，为了方便交流，埃文·威廉姆斯偶然发明了博客。后来，博客变成了公司的主业，博客网站（Blogger.com）也随之上线。前两年大火特火的博客一词，就是埃文·威廉姆斯发明的。2001 年，《财富》杂志曾将“博客网”评选为最佳创新网站。2000 年，互联网泡沫破灭，埃文·威廉姆斯和他的公司也未能幸免。他别无选择，只有裁掉公司所有员工，最后只剩下他一人独自苦撑。不过，“博客网”不仅改变了网络世界，也改变了埃文·威廉姆斯的人生。2003 年，谷歌公司决定收购“博客网”，

濒临绝境的埃文·威廉姆斯把公司卖给了谷歌，这笔交易让威廉姆斯捞到了他的第一桶金。直到今天，埃文·威廉姆斯仍然认为，“博客网”是他最大的创造。

2005 年，埃文·威廉姆斯成立了一家播客公司，又“偶然”地创造了“Twitter”，谱写了新的互联网奇迹。



从埃文·威廉姆斯身上，我们学到了三点经验：一是积极地思考，创新地工作，想办法让工作更高效、更便捷、更快速。或许，你就发现了机会。二是不断尝试才能创造价值，坚持才能收获成功。三是专注于为用户创造最大的价值，而不是其他，如金钱等。

第10堂课

字符串高级处理技术

（视频讲解：106 分钟）

字符串的操作，主要是使用 PHP 的内置 STRING 函数库，通过它可以实现对字符串的各种操作，是 Web 程序开发不可缺少的内容之一。PHP 程序员必须熟练地掌握字符串的处理技术，才能编写出更加实用、完善的 Web 程序。在本堂课中将对字符串的常用技术进行详细讲解。在新技术层出不穷的今天，能让人难忘的和称得上是伟大的却寥寥无几，其中一定会有正则表达式。然而，最容易被人忽略和让人遗忘的也是正则表达式。一方面，几乎所有的编程语言和文本编辑工具都支持正则表达式。而另一方面，关于正则表达式的书籍、资料却少之又少。本堂课虽然对正则表达式的讲解并不全面，但是，笔者相信常用才是硬道理。

学习摘要：

- 初识字符串
- 转义、还原字符串
- 截取字符串
- 分割、合成字符串
- 替换字符串
- 检索字符串
- 去掉字符串首尾空格和特殊字符
- 字符串与 HTML 转换
- 正则表达式
- PCRE 兼容正则表达式函数

10.1 初识字符串

所谓的字符串是由零个或多个字符构成的一个集合。字符包含以下几种类型。

☑ 数字类型，如 1、2、3 等。

☑ 字母类型，如 a、b、c、d 等。

☑ 特殊字符，如#、$、%、^、&等。

☑ 不可见字符，如\n（换行符）、\r（回车符）、\t（Tab 字符）等。

其中，不可见字符是比较特殊的一组字符，是用来控制字符串格式化输出的，在浏览器上不可见，只能看到字符串输出的结果。

例如，通过 echo 语句输出一组字符串，代码如下：

```
<?php
    echo  "PHP 从入门到精通\rASP 从入门到精通\nJSP 程序开发范例宝典\tPHP 函数参考大全"; //输出字符串
?>
```

在 IE 浏览器中不能直接查看到字符串的运行结果，只有通过“查看源文件”才能看到不可见字符串的运行结果。

运行结果为：PHP 从入门到精通

ASP 从入门到精通

JSP 程序开发范例宝典　PHP 函数参考大全

10.2 转义、还原字符串

在 PHP 编程的过程中，将数据插入到数据库中时可能会引起一些问题，如出现错误或者乱码等，因为数据库将传入的数据中的字符解释成控制符。针对这样的问题，需要对特殊的字符进行转义。

因此，在 PHP 语言中提供了专门处理这些问题的技术，通过 addslashes()函数和 stripslashes()函数转义和还原字符串。

addslashes()函数用来给字符串 str 加入斜线“\”，对指定字符串中的字符进行转义。它可以转义的字符包括单引号“'”、双引号“"”、反斜杠“\”、null 字符“0”。该函数的语法如下：

```
string addslashes ( string str)
```

参数 str 为将要被操作的字符串。

提示：addslashes()函数常用的地方就是在生成 SQL 语句时，对 SQL 语句中的部分字符进行转义。

既然有转义，就应该有还原。stripslashes()函数将 addslashes()函数转义后的字符串 str 还原。stripslashes()函数的语法如下：

```
string stripslashes(string str);
```

参数 str 为将要被操作的字符串。

例如，应用 addcslashes()函数对字符串进行转义，然后应用 stripslashes()函数进行还原，代码如下：

```
<?php
 $str = "select * from tb_book where bookname = 'PHP 编程宝典'";
 $a = addslashes($str);                    //对字符串中的特殊字符进行转义
 echo $a."<br>";                           //输出转义后的字符
```

```
 $b = stripslashes($a);                          //对转义后的字符进行还原
 echo $b."<br>";                                 //将字符原义输出
?>
```

运行结果为：select * from tb_book where bookname = \'PHP 编程宝典\'

select * from tb_book where bookname = 'PHP 编程宝典'

技巧：所有数据在插入数据库之前，有必要应用 addslashes()函数进行字符串转义，以免特殊字符未经转义在插入数据库时出现错误。另外，对于应用 addslashes()函数实现的自动转义字符串可以应用 stripslashes()函数进行还原，但数据在插入数据库之前必须再次进行转义。

你问我答：如何控制转义、还原字符串的范围？

通过 addcslashes()函数和 stripcslashes()函数可以对指定范围内的字符串进行转义、还原。

（1）addcslashes()函数，对指定字符串中的字符进行转义，即在指定的字符 charlist 前加上反斜线。通过该函数可以将要添加到数据库中的字符串进行转义，从而避免出现乱码等问题。该函数的语法如下：

```
string addcslashes ( string str, string charlist)
```

其中，参数 str 为将要被操作的字符串；参数 charlist 指定在字符串中的哪些字符前加上反斜线“\”，如果参数 charlist 中包含有\n、\r 等字符，将以 C 语言风格转换，而其他非字母数字且 ASCII 码低于 32 位以及高于 126 位的字符均转换成使用八进制表示。

（2）stripcslashes()函数实现对 addcslashes()函数转义的字符串 str 进行还原。该函数的语法如下：

```
string stripcslashes ( string str)
```

其中，参数 str 为将要被操作的字符串。

10.3　截取字符串

在 PHP 中对字符串进行截取应用 substr()函数。对字符串进行截取是一个最为常用的方法。

substr()函数从字符串中按照指定位置截取一定长度的字符。如果使用一个正数作为子串起点来调用这个函数，将得到从起点到字符串结束的这个字符串；如果使用一个负数作为子串起点来调用，将得到一个原字符串尾部的一个子串，字符个数等于给定负数的绝对值。该函数的语法如下：

```
string substr ( string str, int start [, int length])
```

其中，参数 str 用来指定字符串对象；参数 start 用来指定开始截取字符串的位置，如果参数 start 为负数，则从字符串的末尾开始截取；参数 length 为可选项，指定截取字符的个数，如果 length 为负数，则表示取到倒数第 length 个字符。

substr()函数的操作流程如图 10.1 所示。

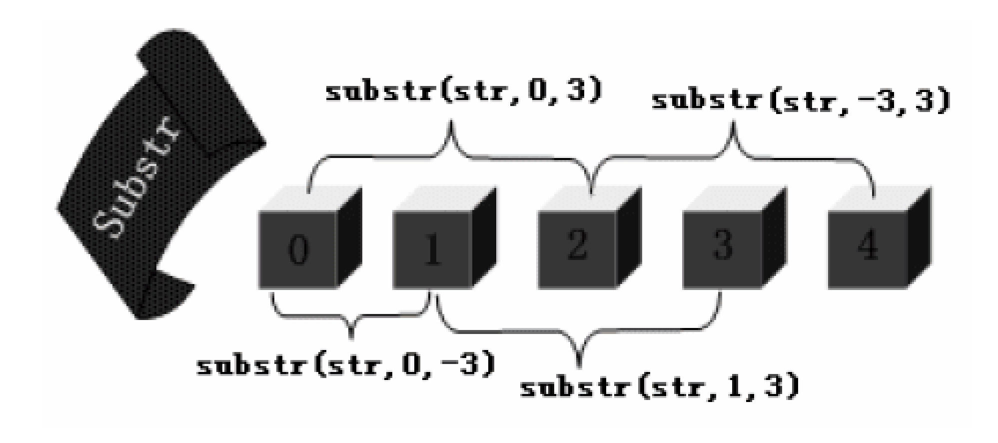


图 10.1　substr()函数的操作流程

注意：substr()函数中参数 start 的指定位置是从 0 开始计算的，即字符串中的第一个字符表示为 0。

例如，在开发 Web 程序时，为了保持整个页面的合理布局，经常需要对一些（如公告标题、公告内容、文章的标题、文章的内容等）超长输出的字符串内容进行截取，并通过“…”代替省略内容。代码如下：

```
<?php
      $str="为进一步丰富编程词典的内容和观赏性，公司决定组织“春季盎然杯”摄影大赛，本次参赛作品要求全部为春季拍摄，旨在展示我国北方地区春季生机盎然的景色。";
      if(strlen($str)>70){                         //如果文本的字符串长度大于 70
            echo substr($str,0,70)."…";            //输出文本的前 70 个字符串，然后输出省略号
}else{                                             //如果文本的字符串长度小于 70
            echo $str;                             //直接输出文本
      }
?>
```

运行结果为：为进一步丰富编程词典的内容和观赏性，公司决定组织“春季盎然杯”摄影大赛…

你问我答：如何控制转义、还原字符串的范围？

strlen()函数获取字符串的长度，汉字占两个字符，数字、英文、小数点、下划线和空格占一个字符。通过 strlen()函数还可以检测字符串长度。例如，在用户注册中，通过 strlen()函数获取用户填写用户名的长度，然后判断用户名长度是否符合指定的标准。关键代码如下：

```
<?php
if(strlen($_POST["pwd"])<6){                   //检测用户密码的长度是否小于 6，弹出警告信息
      echo "<script>alert('用户密码的长度不得少于 6 位!请重新输入'); history.back();</script>";
}else{                                          //用户密码大于等于 6 位，则弹出该提示信息
      echo "用户信息输入合法！";
}
?>
```

10.4　分割、合成字符串

分割字符串将指定字符串中的内容按照某个规则进行分类存储，进而实现更多的功能。例如，在电子商务网站的购物车中，可以通过特殊标识符“@”将购买的多种商品组合成一个字符串存储在数据表中，在显示购物车中的商品时，通过以“@”作为分割的标识符进行拆分，将商品字符串分割成 N 个数组元素，最后通过 for 循环语句输出数组元素，即输出购买的商品。

字符串的分割使用 explode()函数，按照指定的规则对一个字符串进行分割，返回值为数组。该函数的语法如下：

```
array explode(string separator,string str,[int limit])
```

explode()函数的参数说明如表 10.1 所示。

表 10.1　explode()函数的参数说明

参　数	说　明
separator	必要参数，指定的分割符。如果 separator 为空字符串（""），explode()将返回 FALSE。如果 separator 所包含的值在 str 中找不到，那么 explode()函数将返回包含 str 单个元素的数组
str	必要参数，指定将要被进行分割的字符串

续表

参　　数	说　　明
limit	可选参数，如果设置了 limit 参数，则返回的数组包含最多 limit 个元素，而最后的元素将包含 string 的剩余部分；如果 limit 参数是负数，则返回除了最后的-limit 个元素外的所有元素

例如，应用 explode()函数对指定的字符串以@为分隔符进行拆分，并输出返回的数组。代码如下：

```
<?php
    $str="PHP 编程宝典@NET 编程宝典@ASP 编程宝典@JSP 编程宝典";          //定义字符串常量
    $str_arr=explode("@",$str);                                    //应用标识@分割字符串
    print_r($str_arr);                                             //输出字符串分割后的结果
?>
```

运行结果为：Array ([0] => PHP 编程宝典 [1] => NET 编程宝典 [2] => ASP 编程宝典 [3] => JSP 编程宝典)

你问我答：如何将数组中的元素合成字符串？

既然可以对字符串进行分割，返回数组，那么就一定可以对数组进行合成，返回一个字符串。这就是 implode()函数，将数组中的元素组合成一个新字符串。该函数的语法如下：

```
string implode(string glue, array pieces)
```

其中，参数 glue 是字符串类型，指定分隔符；参数 pieces 是数组类型，指定要被合并的数组。

例如，应用 implode()函数将数组中的内容以*为分隔符进行连接，从而组合成一个新的字符串。代码如下：

```
<?php
$str="PHP 编程宝典*NET 编程宝典*ASP 编程宝典*JSP 编程宝典";          //定义字符串常量
$str_arr=explode("*",$str);                                    //应用标识*分割字符串
$array=implode("*",$str_arr);                                  //将数组合成字符串
echo $array;                                                   //输出字符串
?>
```

运行结果为：PHP 编程宝典*NET 编程宝典*ASP 编程宝典*JSP 编程宝典

10.5　替换字符串

字符串的替换技术，可以屏蔽帖子或者留言板中的非法字符，可以对查询的关键字进行描红。PHP 中提供 str_ireplace()函数和 substr_replace()函数实现字符串的替换功能。

10.5.1　str_ireplace()函数

str_ireplace()函数使用新的子字符串（子串）替换原始字符串中被指定要替换的字符串。其语法如下：

```
mixed str_ireplace ( mixed search, mixed replace, mixed subject [, int &count])
```

将所有在参数 subject 中出现的参数 search 以参数 replace 取代，参数&count 表示取代字符串执行的次数。

str_ireplace()函数的参数说明如表 10.2 所示。

表 10.2　str_ireplace()函数的参数说明

参　　数	说　　明
search	必要参数，指定需要查找的字符串
replace	必要参数，指定替换的值
subject	必要参数，指定查找的范围
count	可选参数，获取执行替换的数量

例如，应用 str_ireplace()函数将文本中的字符串“某某”替换为“吉林省明日科技”。代码如下：

```
<?php
$str="    MRSOFT 公司是一家以计算机软件技术为核心的高科技企业，多年来始终致力于行业管理软件开发、数字化出版物制作、计算机网络系统综合应用以及行业电子商务网站开发等领域，涉及生产、管理、控制、仓贮、物流、营销、服务等行业";
        //定义字符串常量
echo str_ireplace("mrsoft","吉林省明日科技",$str);                    //输出替换后的字符串
?>
```

运行结果为：吉林省明日科技公司是一家以计算机软件技术为核心的高科技企业，多年来始终致力于行业管理软件开发、数字化出版物制作、计算机网络系统综合应用以及行业电子商务网站开发等领域，涉及生产、管理、控制、仓贮、物流、营销、服务等行业

注意：本函数不区分大小写。如果需要对大小写加以区分，可以使用 str_replace()函数。

你问我答：关键字描红技术。

关键字描红是指将查询关键字以特殊的颜色、字号或字体进行标识。这样可以使浏览者快速检索到所需的关键字，方便浏览者从搜索结果中查找所需内容。查询关键字描红适用于模糊查询。

10.5.2　substr_replace()函数

substr_replace()函数对指定字符串中的部分字符串进行替换。其语法如下：

```
string substr_replace(string str,string repl,int start,[int length])
```

substr_replace()函数的参数说明如表 10.3 所示。

表 10.3　substr_replace()函数的参数说明

参　　数	说　　明
str	指定要操作的原始字符串
repl	指定替换后的新字符串
start	指定替换字符串开始的位置。正数表示起始位置从字符串开头开始，负数表示起始位置从字符串的结尾开始，0 表示起始位置从字符串中的第一个字符开始
length	可选参数，指定返回的字符串长度。默认值是整个字符串。正数表示起始位置从字符串开头开始，负数表示起始位置从字符串的结尾开始，0 表示“插入”非“替代”

说明：如果参数 start 设置为负数，而参数 length 数值小于或等于 start 数值，那么 length 的值自动为 0。

例如，使用 substr_replace()函数对指定字符串进行替换。代码如下：

```
<?php
$str="用今日的辛勤工作，换明日的双倍回报！";                    //定义字符串常量
```



```
$replace="百倍";                                          //定义要替换的字符串
echo substr_replace($str,$replace,26,4);                  //替换字符串
?>
```

在上面的代码中，使用 substr_replace()函数将字符串“双倍”替换为字符串“百倍”。

运行结果为：用今日的辛勤工作，换明日的百倍回报！

10.6　检索字符串

在 PHP 中，提供了很多应用于字符串查找的函数，如 strstr()函数和 substr_count()函数。PHP 也可以像 Word 那样实现对字符串的查找功能。

10.6.1　strstr()函数

strstr()函数获取一个指定字符串在另一个字符串中首次出现的位置到后者末尾的子字符串。如果执行成功，则返回剩余字符串（存在相匹配的字符）；否则返回 FALSE。该函数的语法如下：

```
string strstr ( string haystack, string needle)
```

其中，参数 haystack 指定从哪个字符串中进行搜索；参数 needle 指定搜索的对象。如果该参数是一个数值，那么将搜索与这个数值的 ASCII 码相匹配的字符。

你问我答：检索字符串函数扩展。

strstr()函数区分大小写，如果不需要对大小写加以区分，可以使用 stristr()函数。

strstr()函数从指定字符在另一个字符串中首次出现的位置开始查找；如果想从指定字符在另一个字符串中最后一次出现的位置开始查找，则可以使用 strrchr()函数。该函数区分大小写。

stripos()函数查找指定字符串（A）在另一个字符串（B）中首次出现的位置。该函数不区分大小写。如果要区分大小写，可以使用 strpos()函数。

strripos()函数查找指定字符串（A）在另一个字符串（B）中最后一次出现的位置。该函数不区分大小写。如果要区分大小写，可以使用 strrpos()函数。

10.6.2　substr_count()函数

10.6.1 小节中介绍的检索字符串的函数，都是检索的指定字符串在另一字符串中出现的位置，这里再介绍一个检索子串在字符串中出现次数的函数——substr_count()。

substr_count()函数获取子串在字符串中出现的次数。其语法如下：

```
int substr_count(string haystack,string needle)
```

其中，参数 haystack 是指定的字符串，参数 needle 为指定的子串。

例如，使用 substr_count()函数获取子串在字符串中出现的次数。代码如下：

```
<?php
$str="PHP 编程宝典、JavaWeb 编程宝典、Java 编程宝典、VB 编程宝典";        //输出查询的字符串
echo substr_count($str,"编程宝典");                                      //输出查询的字符串
?>
```

运行结果为：4

注意：检索子串出现的次数一般常用于搜索引擎中，针对子串在字符串中出现的次数进行统计，便于用户第一时间掌握子串在字符串中出现的次数。

10.7 去掉字符串首尾空格和特殊字符

用户在输入数据时，经常会在无意中输入多余的空白字符，在有些情况下，字符串中不允许出现空白字符和特殊字符，这就需要去除字符串中的空白字符和特殊字符。在 PHP 中提供了 trim()函数去除字符串左右两边的空白字符和特殊字符、ltrim()函数去除字符串左边的空白字符和特殊字符、rtrim()函数去除字符串中右边的空白字符和特殊字符。

10.7.1 ltrim()函数

ltrim()函数用于去除字符串左边的空白字符或者指定字符串。其语法如下：

```
string ltrim( string str [,string charlist]);
```

其中，参数 str 是要操作的字符串对象；参数 charlist 为可选参数，指定需要从指定的字符串中删除哪些字符，如果不设置该参数，则所有的可选字符都将被删除。参数 charlist 的可选值如表 10.4 所示。

表 10.4 ltrim()函数的参数 charlist 的可选值

参　数　值	说　　明
\0	null，空值
\t	Tab，制表符
\n	换行符
\x0B	垂直制表符
\r	回车符
" "	空白字符

注意：除了以上默认的过滤字符列表外，也可以在 Charlist 参数中提供要过滤的特殊字符。

例如，使用 ltrim()函数去除字符串左边的空白字符及特殊字符“(:*_*”。代码如下：

```
<?php
$str="   (:@_@   有一条路走过了总会想起！   @_@:)    ";
$strs="   (:*_*   有一条路走过了总会想起！   *_*:)    ";
echo $str."\n";                          //输出原始字符串
echo ltrim($str)."\n";                   //去除字符串左边的空白字符
echo $strs."\n";                         //输出原始字符串
echo ltrim($strs,"(:*_* ");              //去除字符串左边的特殊字符(:*_*
?>
```

查看源文件的运行结果如图 10.2 所示。

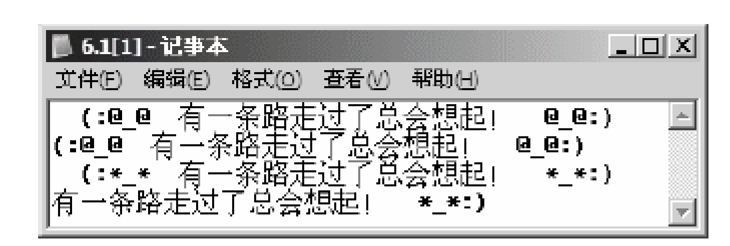


图 10.2 去除字符串左边的空白字符及特殊字符



10.7.2　rtrim()函数

rtrim()函数用于去除字符串右边的空白字符和特殊字符。其语法如下：

```
String rtrim(string str [,string charlist]);
```

其中，参数 str 指定操作的字符串对象，参数 charlist 为可选参数，指定需要从指定的字符串中删除哪些字符，如果不设置该参数，则所有的可选字符都将被删除。参数 charlist 的可选值如表 10.4 所示。

例如，使用 rtrim()函数去除字符串右边的空白字符及特殊字符“(:*_*”。代码如下：

```
<?php
$str="   (:@_@   有一条路走过了总会想起！    @_@:)     ";
$strs="   (:*_*   有一条路走过了总会想起！    *_*:)     ";
echo $str."\n";                                  //输出原始字符串
echo rtrim($str)."\n";                           //去除字符串右边的空白字符
echo $strs."\n";                                 //输出原始字符串
echo rtrim($strs,"    *_*:)");                   //去除字符串右边的特殊字符(:*_*
?>
```

查看源文件的运行结果如图 10.3 所示。

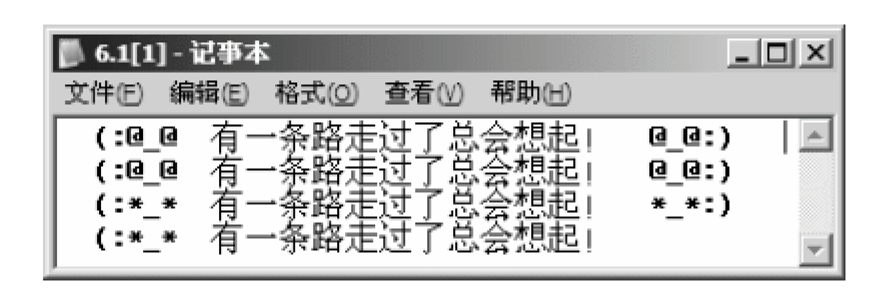


图 10.3　去除字符串右边的空白字符及特殊字符

10.7.3　trim()函数

trim()函数用于去除字符串开始位置和结束位置的空白字符，并返回去掉空白字符后的字符串。该函数的语法如下：

```
string trim(string str [,string charlist]);
```

其中，参数 str 是操作的字符串对象，参数 charlist 为可选参数，指定需要从指定的字符串中删除哪些字符，如果不设置该参数，则所有的可选字符都将被删除。参数 charlist 的可选值如表 10.4 所示。

例如，使用 trim()函数去除字符串左右两边的空白字符及特殊字符“\r\r(: :)”。代码如下：

```
<?php
$str="      \r\r(:@_@去除字符串左右两边的空白和特殊字符  @_@:)          ";
echo $str."\n";                             //输出原始字符串
echo trim($str)."\n";                       //去除字符串左右两边的空白字符
echo trim($str,"\r\r(:@_@ @_@:)");  //去除字符串左右两边的空白字符和特殊字符\r\r(:@_@ @_@:)
?>
```

查看源文件的运行结果如图 10.4 所示。

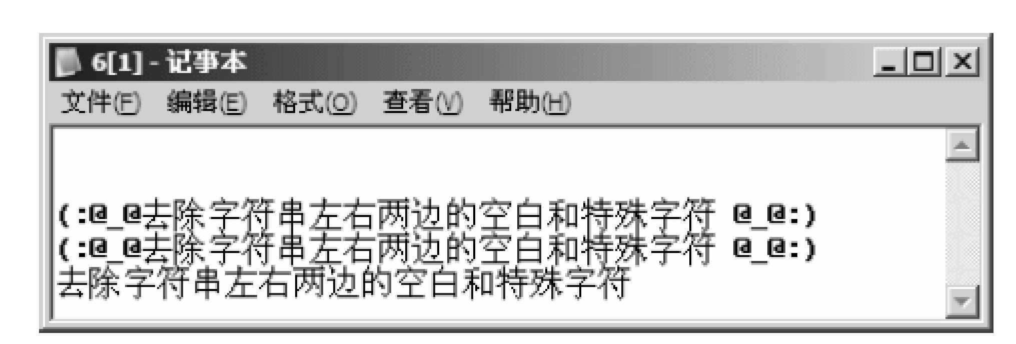


图 10.4　trim()函数去除字符串左右的空格和特殊字符

10.8　字符串与 HTML 转换

字符串与 HTML 之间的转换直接将源代码在网页中输出，而不被执行。这个操作应用最多的地方就是在论坛或者博客的帖子输出中，通过转换直接将提交的源码输出，而确保源码不被解析。完成这个操作主要应用 htmlentities()函数。

htmlentities()函数将所有的字符都转成 HTML 字符串。其语法如下：

```
string htmlentities(string string,[int quote_style],[string charset])
```

htmlentities()函数的参数说明如表 10.5 所示。

表 10.5　htmlentities()函数的参数说明

参　数	说　明
string	必要参数，指定要转换的字符串
quote_style	可选参数，选择如何处理字符串中的引号，有 3 个可选值：（1）ENT_COMPAT，转换双引号，忽略单引号，它是默认值；（2）ENT_NOQUOTES，忽略双引号和单引号；（3）ENT_QUOTES，转换双引号和单引号
charset	可选参数，确定转换所使用的字符集，默认字符集是 ISO-8859-1，指定字符集后就能够避免转换中文字符出现乱码的问题

htmlentities()函数支持的字符集如表 10.6 所示。

表 10.6　htmlentities()函数支持的字符集

字　符　集	说　明
BIG5	繁体中文
BIG5-HKSCS	香港扩展的 BIG5，繁体中文
cp866	DOS 特有的西里尔（Cyrillic）字符集
cp1251	Windows 特有的西里尔字符集
cp1252	Windows 特有的西欧字符集
EUC-JP	日文
GB2312	简体中文
ISO-8859-1	西欧，Latin-1
ISO-8859-15	西欧，Latin-9
KOI8-R	俄语
Shift-JIS	日文
UTF-8	ASCII 兼容的多字节 8 编码

10.9　正则表达式

10.9.1　什么是正则表达式

正则表达式是一种描述字符串结构的语法规则，是一个特定的格式化模式，可以匹配、替换、截取匹



配的字串。对于用户来说，在注册某个网站的会员时，都提交过邮箱、电话号码或者邮编等注册信息，但是当填写的注册信息的格式不正确时就会弹出一个提示信息，提示您填写的邮箱或者电话号码的格式不正确，那么这个判断的依据就是正则表达式，通过正则表达式来对您提交的信息进行匹配，如果不匹配那么就将给出提示信息。

例如，匹配邮箱和电话号码格式的正则表达式如下：

```
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/                //验证邮箱格式是否正确
/^(\d{3}-)(\d{8})$|^(\d{4}-)(\d{7})$|^(\d{4}-)(\d{8})$/;      //验证座机号码
```

如果要应用正则表达式，必须将其放置到某种语言中，通过语言中的方法调用正则表达式对提交的信息进行验证。在 PHP 中有两套函数库支持正则表达式的处理操作。一套是由 PCRE（Perl Compatible Regular Expression）库提供的，与 Perl 语言兼容的正则表达式函数，其以“preg_”为前缀命名函数，而且表达式要包含在定界符“/”之内。另一套是 POSIX（Portable Operating System Interface of Vnix）扩展语法正则表达式函数，其以“ereg_”为前缀命名函数。两套函数库的功能非常相似，但是在执行效率上 PCRE 函数库要略优于 POSIX 函数库。

你问我答：grep、egrep、POSIX、Perl 和 PCRE。

grep：最初是 ED 编辑器中的一条命令，用来显示文件中特定的内容。后来成为一个独立的工具 grep。

egrep：grep 虽然不断地更新升级。但仍然无法跟上技术的脚步。为此，贝尔实验室推出了 egrep，意为“扩展的 grep”。这大大增强了正则表达式的能力。

POSIX（Portable Operating System Interface of Vnix）：可移植操作系统接口。在 grep 发展的同时，其他一些开发人员也按照自己的喜好开发出了具有独特风格的版本。但问题也随之而来，有的程序支持某个元字符，而有的程序则不支持。因此就有了 POSIX，POSIX 是一系列标准，确保了操作系统之间的可移植性。但 POSIX 和 SQL 一样，没有成为最终的标准而只能作为一个参考。

Perl（Practical Extraction and Reporting Language）：实际抽取与汇报语言。1987 年，Larry Wall 发布了 Perl。在随后的 7 年时间里，Perl 经历了从 Perl 1 到现在的 Perl 5 的发展。最终 Perl 成为了 POSIX 之后的另一个标准。

PCRE：Perl 的成功，让其他的开发人员在某种程度上要兼容 Perl，包括 C/C++、Java、Python 等都有自己的正则表达式。1997 年，Philip Hazel 开发了 PCRE 库，这是兼容 Perl 正则表达式的一套正则引擎，其他开发人员可以将 PCRE 整合到自己的语言中，为用户提供丰富的正则功能。许多软件都使用 PCRE，PHP 正是其中之一。

10.9.2　正则表达式语法规则

一个完整的正则表达式由两部分构成，即元字符和文本字符。元字符就是具有特殊含义的字符，如前面提到的“*”和“？”。文本字符就是普通的文本，如字母和数字等。PCRE 风格的正则表达式一般都放置在定界符“/”中间，如“/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/”、“/^http:\/\/(www\.)?.+.?$/”。

为了便于读者理解，除了个别字符外，元字符的相关说明和分类如表 10.7 所示。

表 10.7　元字符的相关说明和分类

元　字　符	说　　明	举　　例
行定位符（^和$）	行定位符就是用来描述字串的边界	^tm
单词定界符（\b、\B）	表达式 tm 可以对字符串中任意位置出现的字串 tm 进行匹配	\btm\b

续表

元　字　符	说　　明	举　　例
字符类（[ ]）	方括号内不区分大小写	[Tt][Mm]
选择字符（|）	该字符可以理解为“或”，与“[ ]”相同	(T|t)(M|m)
连字符（-）	变量的命名规则是只能以字母和下划线开头	[a-zA-Z]
排除字符（[^]）	匹配不符合命名规则的变量	[^a-zA-Z]
限定符（? * + {n,m}）	对于重复出现的字母或字串，可以使用限定符来实现匹配	go{2,20}gle
点号字符（.）	匹配除换行符外的任意一个字符	^s.t$
转义字符（\）	正则表达式中的转移字符（\）和 PHP 中的大同小异，都是将特殊字符转义为普通字符	[0-9]{1,3}(\.[0-9]{1,3}){3}
括号字符（()）	改变限定符的作用范围	(thir|four)th

10.10　PCRE 兼容正则表达式函数

PHP 中提供了两套支持正则表达式的函数库，但是由于 PCRE 函数库在执行效率上要略优于 POSIX 函数库，所以这里只讲解 PCRE 函数库中的函数。在 PCRE 函数库中包含 preg_grep()、preg_match()、preg_match_all()、preg_quote()、preg_replace()、preg_replace_callback()和 preg_split()7 个函数。

10.10.1　字符串的匹配与查找

在 PCRE 函数库中，通过 preg_match()函数、preg_match_all()函数和 preg_grep()函数完成复杂的字符串匹配与查找操作。

1．preg_match()函数

preg_match()函数，根据正则表达式的模式对指定的字符串进行搜索和匹配。其语法如下：

```
int preg_match ( string pattern, string subject [, array matches [, int flags]] )
```

参数说明：

☑ pattern：定义正则表达式的模式。

☑ subject：指定被搜索的字符串。

☑ matches：可选参数，存储与第一个参数中的子模式的各个部分的匹配结果。

你问我答：子模式。

正则表达式中的子模式是使用括号“()”括起的模式单元，数组 matches 中的第一个元素 matches[0]保存与正则表达式 pattern 匹配的整体内容，而数组 matches 中的其他元素则按照顺序保存了与正则表达式小括号内子表达式相匹配的内容。

☑ flags：其标记是 PREG_OFFSET_CAPTURE，如果设定本标记，对每个出现的匹配结果也同时返回其附属的字符串偏移量。注意这改变了返回的数组的值，使其中的每个单元也是一个数组，其中第一项为匹配字符串，第二项为其偏移量。本标记自 PHP 4.3.0 起可用。

2. preg_match_all()函数

preg_match_all()函数与 preg_match()函数类似，都是根据正则表达式的模式对指定的字符串进行搜索和匹配。其区别是 preg_match()函数在第一次匹配成功后就停止查找，而 preg_match_all()函数则会一直匹配到最后才会停止，获取到所有相匹配的结果。其语法如下：

```
int preg_match_all ( string pattern, string subject, array matches [, int flags] )
```

preg_match_all()函数的参数说明如表 10.8 所示。

表 10.8　preg_match_all()函数的参数说明

参　　数	说　　明
pattern	定义正则表达式的模式
subject	指定被搜索的字符串
matches	可选参数，存储所有的匹配结果，并且返回整个模式匹配的次数，如果出错则返回 FALSE
flags	可选参数，根据其指定的顺序将每次匹配的结果存储到第 3 个参数中。参数 flag 的可选值包括 PREG_PATTERN_ORDER 和 PREG_SET_ORDER。 ☑ PREG_PATTERN_ORDER：默认排序方式，数组 matches 中的第一个元素 matches[0]保存与正则表达式 pattern 匹配的整体内容，而数组 matches 中的其他元素则按照顺序保存与正则表达式小括号内子表达式相匹配的内容 ☑ PREG_SET_ORDER：确立 matches[0]元素中存储第一组匹配的数组，matches[1]元素中存储第二组匹配的数组，依此类推

3. preg_grep()函数

preg_grep()函数对数组中的元素进行匹配。其语法如下：

```
array preg_grep ( string pattern, array input )
```

其中，参数 pattern 定义正则表达式的模式；参数 input 指定被搜索的数组。

该函数返回一个数组，其中包含了第二个参数 input 数组中与给定的第一个参数 pattern 模式相匹配的单元，preg_grep()函数对于输入数组 input 中的每个元素只进行一次匹配。

你问我答：字符串处理方式的选择。

虽然通过正则表达式能够很好地完成字符串的匹配与查找工作，但是在使用时应该做到物尽其用，在完成一些复杂的字符串处理时使用正则表达式。

而在完成一些字符串的简单操作时，还是首先从字符串函数库中去考虑。例如，查找一个字符串中是否包含某个子字符串，可以使用 strstr()或者 strpos()函数；完成对一个字符串的截取操作，可以使用 substr()函数；获取指定字符在字符串中出现的次数，可以使用 substr_count()函数。

毕竟使用字符串函数库中的函数要比使用正则表达式的执行效率要高一些。

10.10.2　字符串的替换

在 PCRE 函数库中，通过 preg_replace()函数和 preg_replace_callback()函数完成复杂的字符串替换操作。同样对于简单的字符串替换操作，可以使用字符串函数库中的 str_replace()函数来完成。

1. preg_replace()函数

preg_replace()函数，根据正则表达式的模式完成字符串的搜索和替换操作。其语法如下：

```
mixed preg_replace ( mixed pattern, mixed replacement, mixed subject [, int limit] )
```

preg_replace()函数的参数说明如表 10.9 所示。

表 10.9 preg_replace()函数的参数说明

参　数	说　明
pattern	定义正则表达式的模式
subject	指定被搜索的字符串
replacement	指定替换的内容
limit	可选参数，指定替换的次数。如果省略 limit 或者其值为-1，则所有的匹配项都会被替换

函数功能：该函数在字符串 subject 中匹配表达式 pattern，并将匹配项替换成字串 replacement。如果有参数 limit，则替换 limit 次。

注意：preg_replace()的每个参数（除了 limit）都可以是一个数组。如果 pattern 和 replacement 都是数组，将以其键名在数组中出现的顺序来进行处理。这不一定和索引的数字顺序相同。如果使用索引来标识哪个 pattern 将被哪个 replacement 替换，那么应该在调用 preg_replace()之前用 ksort()函数对数组进行排序。

2. preg_replace_callback()函数

preg_replace_callback 函数与 preg_replace()函数的功能相同，都用于查找和替换字串。不同的是 preg_replace_callback()函数使用一个回调函数（callback）来代替 replacement 参数。其语法如下：

```
mixed preg_replace_callback ( mixed pattern, callback callback, mixed subject [, int limit] )
```

注意：在 preg_replace_callback()函数的回调函数中，建议使用单引号定义字符串，这样可以保证字符串中的特殊符号不被转义。

10.10.3 字符串的分割

在 PCRE 函数库中，通过 preg_split()函数完成复杂的字符串分割操作。但是对于简单的字符串分割操作，则可以使用字符串函数库中的 explode()函数来完成。

preg_split()函数，根据正则表达式定义的模式完成对指定字符串的分割操作。其语法如下：

```
array preg_split ( string pattern, string subject [, int limit [, int flags]] )
```

参数说明：

- ☑ pattern：定义正则表达式的模式。
- ☑ subject：指定被分割的字符串。
- ☑ limit：可选参数，返回最多 limit 个字串，而其中最后一个元素包含 subject 中剩余的所有部分。如果 limit 是-1，则表示没有限制。
- ☑ flags：可选参数，可以是下列标记的任意组合（用按位或运算符 | 组合）。
  - ➢ PREG_SPLIT_NO_EMPTY：如果设定本标记，则 preg_split()只返回非空的成分。
  - ➢ PREG_SPLIT_DELIM_CAPTURE：如果设定本标记，定界符模式中的括号表达式也会被捕获并返回。本标记添加于 PHP 4.0.5。
  - ➢ PREG_SPLIT_OFFSET_CAPTURE：如果设定本标记，对每个出现的匹配结果也同时返回其附属的字符串偏移量。注意这改变了返回的数组的值，使其中的每个单元也是一个数组，其中第一项为匹配字符串，第二项为其在 subject 中的偏移量。本标记自 PHP 4.3.0 起可用。

10.11　照猫画虎——基本功训练

10.11.1　基本功训练 1——在网页中直接输出 HTML 脚本

视频讲解：光盘\mr\10\lx\在网页中直接输出 HTML 脚本.exe

实例位置：光盘\mr\10\zmhh\01

对 HTML 代码格式化输出是十分必要的。因为在默认情况下，程序会解析 HTML 代码生成类似于图 10.5 所示的内容，而用户可能只是单纯地想将 HTML 代码作为一个字符串输出到页面中。本实例可以为用户实现这样一个功能，运行结果如图 10.6 所示。

⊙正常输出　○格式化输出　确定

明日科技			
C#部门	JAVA部门	PHP部门	VB部门
小可	小李	小杨	小巩
小刘	小陈	小潘	小安
小周	小张	小王	小孙

图 10.5　直接输出 HTML 代码

○正常输出　○格式化输出　确定

<table> <thead><td colspan="4" align="center">明日科技</td></thead> <tbody> <tr><td>C#部门</td><td>JAVA部门</td><td>PHP部门</td><td>VB部门</td></tr> <tr><td>小可</td><td>小李</td><td>小杨</td><td>小巩</td></tr> <tr><td>小刘</td><td>小陈</td><td>小潘</td><td>小安</td></tr> <tr><td>小周</td><td>小张</td><td>小王</td><td>小孙</td></tr> </tbody> </table>

图 10.6　格式化输出 HTML 代码

本实例的实现过程非常简单，只需要将编写好的 HTML 表格代码通过 PHP 中的 htmlspecialchars()函数转义输出即可。代码如下：

```
<?php
	if(isset($_POST[sub])){						//当“确定”按钮被单击时
		if($_POST[radio]==""){					//判断单选按钮是否为空
			echo $string;					//输出字符串
		}else{
			$content=htmlspecialchars($string);
			echo $content;					//格式化输出字符串
		}
	}
?>
```

照猫画虎：在实际的开发过程中，用户经常会使用 JavaScript 脚本语言，用于简单的提示，应用上面的原理输出动态变化的数字倒计时。（15 分）（实例位置：光盘\mr\10\zmhh\01_zmhh）

10.11.2　基本功训练 2——substr()函数截取超长字符串

视频讲解：光盘\mr\10\lx\substr()函数截取超长字符串.exe

实例位置：光盘\mr\10\zmhh\02

如果用户只是对英文字符串进行截取，那么学习本训练似乎没有多大的意义。但是如果对中文字符串进行截取，那么本训练的重要性就体现得淋漓尽致。大家都知道一个中文字是由两个字符组成的，如果在截取的过程中只截取了一个字符，那么本中文字符就会体现为乱码输出。本实例可以完美地解决这个问题，运行结果如图 10.7 所示。



图 10.7　截取超长字符串

具体步骤如下：

（1）创建脚本文件 index.php。编写<form>表单，在表单中设定一个文本框和一个提交按钮。

（2）编写自定义函数，此函数判断传入的每一个字符的 ASCII 码是否大于 0xa0。如果大于 0xa0，说明此字符为一个中文字的一部分，按两个字节截取，否则按一个字符截取。代码如下：

```
function str_sub($string,$start,$len){
    $strlen=$start+$len;
    for($a=0;$a<$strlen;$a++){
        if(ord(substr($strlen,$a,1))>0xa0){
            $str.=substr($string,$a,2);
        }else{
            $str.=substr($string,$a,1);
        }
    }
    return $str;
}
```

（3）当“截取”按钮被单击时，将文本框中所有截取的字符长度动态地传入到自定义函数中并输出截取内容。代码如下：

```
if(isset($_POST[sub])){                              //当“截取”按钮被单击时
        if(preg_match("/\d/",$_POST[text])){          //正则表达式判断文本框数据是否为数字
            $len=$_POST[text];
            if(is_file("index")){                    //读取文本文件并输出截取的字符串
                $string=file_get_contents("index");
                echo str_sub($string,0,$len);
                exit;
            }
        }else{
            echo "文本框内容只能是数字";
        }
}
```

照猫画虎：应用 PHP 函数 substr()实现从字符串末尾开始截取指定长度的字符串。（15 分）（实例位置：光盘\mr\10\zmhh\02_zmhh）

10.11.3　基本功训练 3——统一英文注册用户首字母的大小写

视频讲解：光盘\mr\10\lx\统一英文注册用户首字母的大小写.exe

实例位置：光盘\mr\10\zmhh\03

网站的国际化已经不是一个炙手可热的话题，但是由于地区差异在使用英文填写注册信息时首字母大小写得不到统一。本实例利用函数 ucfirst()实现英文注册用户首字母统一为大写，运行结果如图 10.8 所示。

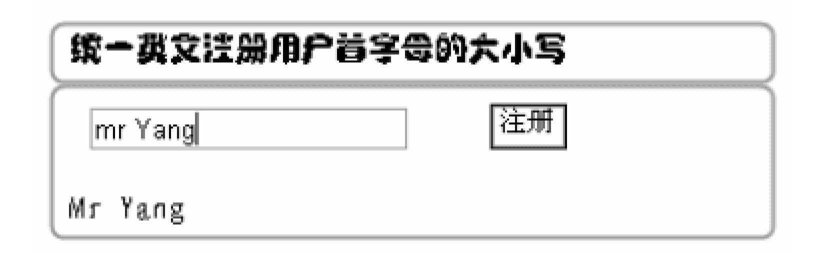


图 10.8　首字母大小写

具体步骤如下：

创建 PHP 脚本，当单击“注册”按钮时，首先，利用$_POST[]全局数组获取从文本框传递进来的数据。然后，将获取的数据经 ucfirst()函数转换为首字母大写。最后，输出经转换后的数据。代码如下：



```
<?php
        if(isset($_POST[sub])){                 //当“注册”按钮被单击时
            echo ucfirst($_POST[text]);       //将首字母转换为大写
        }
    ?>
```

照猫画虎：将英文字符串全部转换为大写。提示：应用 strtoupper()函数。（15 分）（实例位置：光盘\mr\10\zmhh\03_zmhh）

10.11.4　基本功训练 4——货币数据的格式化输出

视频讲解：光盘\mr\10\lx\货币数据的格式化输出.exe

实例位置：光盘\mr\10\zmhh\04

货币数据不同于其他整型或浮点型数据，货币数据是存在一定格式的。本实例通过函数 number_format()实现金额的格式化输出，运行结果如图 10.9 所示。

图 10.9　货币数据的格式化输出

实现本实例的操作非常简单，当单击“注册”按钮时，程序自动将文本框中的信息进行格式化。代码如下：

```
<?php
if(isset($_POST[sub])){
            echo number_format($_POST[text]);
        }
    ?>
```

照猫画虎：将二维数组合成为字符串并实现输出。（15 分）（实例位置：光盘\mr\10\zmhh\04_zmhh）

10.11.5　基本功训练 5——验证 email 地址格式是否正确

视频讲解：光盘\mr\10\lx\验证 email 地址格式是否正确.exe

实例位置：光盘\mr\10\zmhh\05

互联网发展到今天，几乎所有的 Web 爱好者都有自己的 email 地址，无论您申请的是 126 邮箱，还是 163 邮箱，Email 地址的格式是固定的。本实例通过 preg_match()函数和正则表达式验证 email 地址格式是否正确，运行结果如图 10.10 所示。

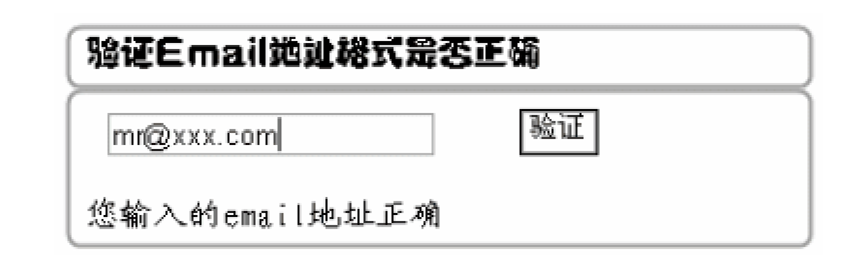


图 10.10　验证 email 地址是否正确

本实例的实现方法非常简单，当“验证”按钮被单击时，将获取的文本框信息作为正则表达式函数的参数实现验证。其中验证 email 的正则表达代码如下：

```
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/
```

程序代码如下：

```
<?php
    if(isset($_POST[sub])){                    //页面中存在 sub 变量
        if(preg_match("/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/",trim($_POST['text']))){//正则表达式验证
            echo "您输入的 email 地址正确";                                  //输出提示
        }else{
            echo "您输入的 email 地址不正确";
        }
```

```
    }
?>
```

照猫画虎：利用正则表达式函数验证邮政编码格式是否正确。（20 分）（实例位置：光盘\mr\10\zmhh\05_zmhh）

10.11.6　基本功训练 6——实现字符串的替换

视频讲解：光盘\mr\10\lx\实现字符串的替换.exe

实例位置：光盘\mr\10\zmhh\06

替换字符串需要使用 str_replace()函数，不要小看这个函数，它应用的范围是很广泛的。例如，很多网站的站内搜索引擎所使用的关键字描红技术等都可以通过此函数来实现。本实例只是简单的字符串之间的替换，并没有将指定的关键字描红。运行结果如图 10.11 所示。

原字符串内容为:
明日科技是一家优秀的IT服务机构
将明日科技替换成mrsoft
mrsoft是一家优秀的IT服务机构

图 10.11　替换指定字符串

本实例利用 str_replace()函数将指定的字符串内容进行替换。其语法格式如下：

```
mixed str_replace ( mixed search, mixed replace, mixed subject [, int &count] )
```

参数说明：

- ☑ mixed search：需要替换的原内容。
- ☑ mixed replace：替换内容。
- ☑ mixed subject：原字符串。

本实例的核心代码如下：

```
<div id="one">
    <div id="two">
    <?php echo "原字符串内容为：";?>
    </div>
    <div id="three">
    <?php echo $string;?>
    </div>
    <div id="four">
    <?php echo "将明日科技替换成 mrsoft";?>
    </div>
    <div id="five">
    <?php
        echo str_replace("明日科技","mrsoft",$string);            //替换操作
    ?>
    </div>
</div>
```

照猫画虎：利用正则表达式函数 preg_replace()实现指定字符串替换。（20 分）（实例位置：光盘\mr\10\zmhh\06_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	6	总分数	
分数								



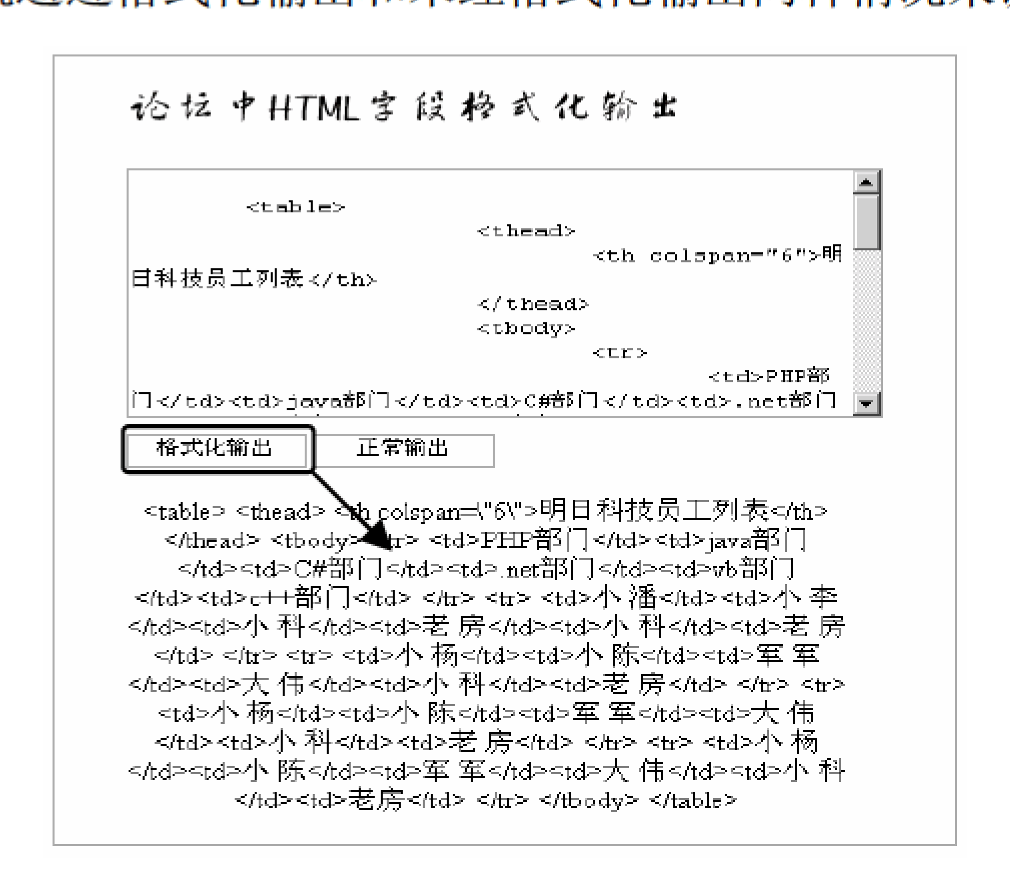
10.12　情景应用——拓展与实践

10.12.1　情景应用 1——在论坛的帖子对 PHP 脚本进行转换输出

视频讲解：光盘\mr\10\lx\在论坛的帖子对 PHP 脚本进行转换输出.exe

实例位置：光盘\mr\10\qjyy\01

用户在编写与浏览者互动的论坛模块中，如发布主题等，一定要对浏览者发布的内容进行 HTML 的格式化输出。这样做的意义非常重大，最主要的原因是，用户论坛的服务器资源是有限的，这样有限的资源去解析浏览者发布的无意或者恶意的 HTML 代码，会给服务器带来莫大的负担甚至是崩溃。为了解决这一问题，下面就通过格式化输出和未经格式化输出两种情况来讲解，制作一个合理的格式化小模块，如图 10.12 所示。



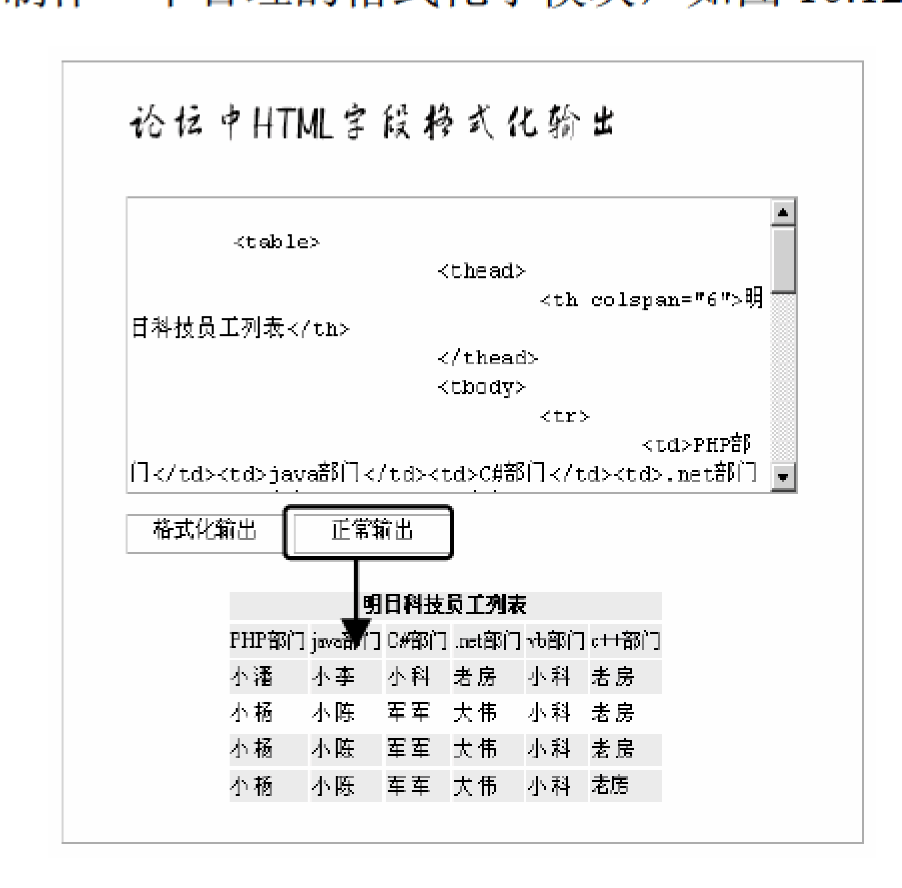


图 10.12　HTML 字段格式化输出

完成本实例，主要是通过自定义函数 html()对用户发布的内容进行格式化。该函数的实现原理是：将传递进来的内容经函数 htmlspecialchars()进行 HTML 代码格式化，这样一来 HTML 代码在网页中就转变成普通的字符串。这样一个简单的格式化输出函数就定义完成了。关键代码如下：

```
<?php
        //自定义函数，content 表示传递进来的内容
        function html($content){
            $content=htmlspecialchars($content);                //将 HTML 代码格式化
            return $content;
        }
?>
```

DIY：利用字符串替换函数替换帖子中的空格符。（15 分）（实例位置：光盘\mr\10\qjyy\01_diy）

10.12.2　情景应用 2——以省略模式输出论坛中最新动态标题

视频讲解：光盘\mr\10\lx\以省略模式输出论坛中最新动态标题.exe

实例位置：光盘\mr\10\qjyy\02

用户在浏览网站时，经常会看到类似于最新动态、热门话题等模块内容。这些模块的主要作用是向浏

览网站的用户提供网站的最新、最热的话题或相关信息的链接，但是这些模块由于主页空间、大小、布局的限制，不可能将所有内容全部显示出来，这就需要程序员对这些热门话题、最新动态的标题进行合理的截取。本实例针对国内的中文网站对该模块的内容标题，进行合理的中文字符串截取。程序运行结果如图 10.13 所示。

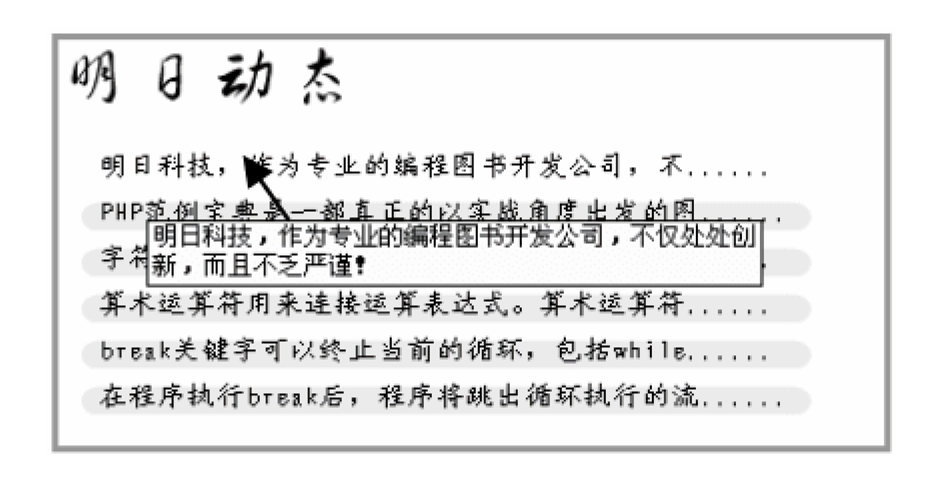


图 10.13　动态标题

完成本实例，主要是通过自定义函数 chinese_substr()对所要截取的字符串内容进行合理的转换截取。在自定义函数中定义 3 个参数：$str 代表截取字符串的内容，$start 参数代表截取字符串的开始位置，$len 代表截取字符串的长度。利用 for 循环语句，将字符串中每一个字节转换成 ASCII 码并与 0xa0 进行比较，当该字节 ASCII 值大于 0xa0 时，表明此字节为中文部分，按两个字节长度截取，否则按一个字节长度截取。代码如下：

```
<?php
	function chinese_substr($str,$start,$len){		//自定义函数
		$strlen=$start+$len;			//截取字符串的总长度
		for($i=0;$i<$strlen;$i++) {		//for 循环语句
			if(ord(substr($str,$i,1))>0xa0) {	//将字节转换成 ASCII 码与 0xa0 比较
				$tmpstr.=substr($str,$i,2);	//按中文字符串截取
				$i++;
			}else{
				$tmpstr.=substr($str,$i,1);	//按单个字节截取
			}
		}
	return $tmpstr;
	}
?>
```

DIY：编写一个中文字符串的截取类用于超长文本的截取重用。（20 分）（实例位置：光盘\mr\10\qjyy\02_diy）

10.12.3　情景应用 3——对查询的关键字进行描红、加粗

视频讲解：光盘\mr\10\lx\对查询的关键字进行描红、加粗.exe

实例位置：光盘\mr\10\qjyy\03

查询关键字描红在百度、谷歌等搜索引擎中能够应用到，通过它可以凸显出要查找的内容。本实例实现的效果如图 10.14 所示。

图 10.14　关键字描红

本实例利用 str_replace()函数完成字符串的替换操作，进而完成对查询关键字的描红。关键代码如下：

```
<?php
		if(isset($_POST[sub])){				//页面中存在 sub 变量
			$dbms='mysql';				//定义连接数据库的相关参数
```

```
            $dbName='db_database10';
            $host='localhost';
            $user='root';
            $pwd='111';
            $dsn="$dbms:host=$host;dbname=$dbName";
            try {                                               //try…catch 捕捉异常
                $pdo=new PDO($dsn,$user,$pwd);                  //实例化对象
                $pdo->query("SET NAMES utf8");                  //设置页面的编码格式
                $sql="select * from tb_demo03 where content like '%".$_POST[text]."%'";
                $rs=$pdo->query($sql);                          //执行查询操作
                $string="<b>".$_POST[text]."</b>";
                foreach($rs as $val){

                    $value=str_replace($_POST[text],$string,$val[1]);   //替换关键字
                    echo "<font>".$value."</font><br>";                 //描红输出
                }
            } catch (Exception $e) {
            }
        }
    ?>
```

DIY：利用 substr_count()函数实现检索关键字在文本中的出现次数。（15 分）（实例位置：光盘\mr\10\qjyy\03_diy）

10.12.4　情景应用 4——通过字符串截取控制图像验证码的输出

视频讲解：光盘\mr\10\lx\通过字符串截取控制图像验证码的输出.exe

实例位置：光盘\mr\10\qjyy\04

图像验证码相信用户并不陌生，它的主要作用是防止用户恶意刷新注册使服务器系统资源流失，数据库冗余信息堆积。验证码的出现很好地解决了或者说是避免了相关问题发生的可能性，创建 PHP 图像验证码需要开启 GD2 函数库支持，利用相关的图像函数创建。本实例的运行结果如图 10.15 所示。

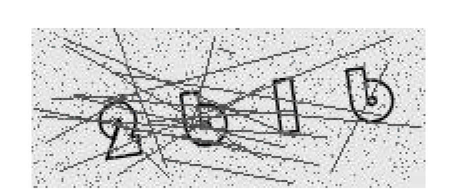


图 10.15　图像验证码

实现本实例主要通过 GD2 函数库的 7 个相关函数，其中包括 imagecreate()、imagecolorallocate()、imagefttext()、imageline()、imagesetpixel()、imagejpeg()和 imagedestroy()控制并生成验证码。然后利用字符串截取函数获取 4 个十六进制数并输出到图像中。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");             //设置页面数据的编码风格
    header("Content-Type:image/jpeg");                          //告知浏览器输出的是个图像
    $image=imagecreate(200,80);                                 //创建画布
    $bgcolor=imagecolorallocate($image,230,230,230);            //设置背景颜色
    $fontcolor=imagecolorallocate($image,0,0,0);                //设置字体颜色
    $linecolor=imagecolorallocate($image,rand(0,200),rand(0,200),rand(0,200));    //设置线的颜色
    $pixelcolor=imagecolorallocate($image,rand(0,200),rand(0,200),rand(0,200));   //设置噪点的颜色
    for($a=0;$a<10;$a++){
        $rand.=dechex(rand(0,15));                              //获取 10 个十六进制数
    }
    for($b=0;$b<4;$b++){
        $str.=substr($rand,-$b,1)."   ";                        //截取 4 个十六进制数
```

```
    }
    $font="STCAIYUN.TTF";                                                    //定义字体
    imagefttext($image,30,rand(0,8),rand(0,40),rand(50,80),$fontcolor,$font,$str);   //输出字符
    for($c=0;$c<20;$c++){
        imageline($image,rand(0,200),rand(0,80),rand(0,200),rand(0,80),$linecolor);  //输出 20 条线
    }
    for($d=0;$d<1000;$d++){
        imagesetpixel($image,rand(0,200),rand(0,80),$pixelcolor);            //输出 1000 个噪点
    }
    imagejpeg($image);                                                       //输出图像
    imagedestroy($image);                                                    //销毁图像
?>
```

DIY：应用上面的实例在验证码图像中添加中文。（15 分）（实例位置：光盘\mr\10\qjyy\04_diy）

10.12.5 情景应用 5——验证上传文件的格式是否正确

视频讲解：光盘\mr\10\lx\验证上传文件的格式是否正确.exe

实例位置：光盘\mr\10\qjyy\05

验证上传文件的格式类型是对上传文件的一种安全保障，大家不妨设想一下，实现任意格式的文件上传一方面给非法用户以可乘之机，另一方面也给服务器带来很大的压力。可见对上传文件格式的规范是很有必要的，本实例将介绍一种控制上传文件格式的方法，运行结果如图 10.16 所示。

验证上传文件的格式

C:\Documents and Setting 浏览... 上传

您上传的文件格式为.doc是程序允许的文件类型

图 10.16 验证上传文件的格式

本实例主要通过两个函数来完成：一是使用字符串截取函数，自定义截取上传文件名称的后 4 位数据；二是利用正则表达式函数 preg_match()实现对指定的字符串进行验证操作。代码如下：

```
<?php
    if(isset($_POST[sub])){                                  //当页面中存在 sub 变量
        $file=$_FILES['filename'];                           //获取上传文件的相关信息
        $arr=array(".jpg",".gif",".txt",".xls",".doc");      //创建数组并添加指定的元素
        $string=implode($arr);                               //将数组合成字符串
        $str=substr($file[name],-4,4);                       //截取上传文件名称的后 4 位
        if(preg_match("/$str/",$string)){                    //利用正则表达式匹配并输出提示信息
            echo "您上传的文件格式为".$str."是程序允许的文件类型";
        }else{
            echo "非法的上传文件类型";
        }
    }
?>
```

DIY：利用 substr()函数获取上传文件的文件类型。（15 分）（实例位置：光盘\mr\10\qjyy\05_diy）

10.12.6 情景应用 6——验证用户提交的数据是否是整数

视频讲解：光盘\mr\10\lx\验证用户提交的数据是否是整数.exe

实例位置：光盘\mr\10\qjyy\06

文本框的验证方式在前面的训练中已经有了初步的接触，像此类的验证方式最好采用 JavaScript 代码实



现验证，以减轻服务器不必要的压力。本实例的运行结果如图 10.17 所示。



图 10.17　验证提交数据

本实例使用正则表达式的匹配函数 preg_match()，将获取到的文本框信息按照指定的格式进行匹配，并输出相关的提示。其中使用正则表达式函数 preg_match()的代码如下：

```
preg_match("/\d/",$_POST[text])
```

本实例的核心代码如下：

```
<?php
    if(isset($_POST[sub])){                                    //当“提交”按钮被单击时
        if(preg_match("/[0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9]/",$_POST[text])){ //正则表达式匹配并输出提示
            echo "<b>您输入的内容是一个数字</b>";
        }else{
            echo "<b>您输入的内容不是一个数字</b>";
        }
    }
?>
```

DIY：利用 PHP 函数判断文本框输入的内容是否为英文字符。（20 分）（实例位置：光盘\mr\10\qjyy\06_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	6	总分数	
分数								

10.13　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．下列（　　）函数是将数组转换为字符串。

A．imploed()　　B．explode()　　C．arsort()　　D．natsort()

2．将字符串中所有英文单词的开头字母转换为大写的函数应该是（　　）。

A．ucfirst()　　B．strtolower()　　C．strtpupper()　　D．ucwords()

3．以下字符的长度是（　　）。

```
<?php
    $text="   \tllo   ";
    Echo strlen(trim($text));
?>
```

A．9　　B．5　　C．7　　D．3

4．PHP 中，下列（　　）函数是将字符串前后颠倒。

A．strrev()　　B．strrpos()　　C．strstr()　　D．strfirst()

5．下面（　　）正则表达式，可以从服务器上取到访问域名第二层。

A．preg_replace（"/.*([^\.\/]+)(\..(com|net|org))?\.[^\.\/]+$/", "\1",$_SERVER['HTTP_HOST']）

B．preg_replace（"/.*?([^\.\/]+)(\..(com|net|org))?\.[^\.\/]+$/", "\1",$_SERVER['HTTP_HOST']）

C．preg_replace（"/.*([^\.\/]+)(\..(com|net|org))?\.[^\.\/]+$/", "\1",$HTTP_HOST）

D．preg_replace（"/.*?([^\.\/]+)(\..(com|net|org))?\.[^\.\/]+$/", "\1",$_SERVER['HTTP_HOST']）

二、填空题（每题 10 分，5 道题）

1．在横线处填写使用的函数。

```
<?php
    $email='mingfisoft@mingrisott.com';
    $str=___①__($email,'@');
    $info=___②__('.',$str);
    ___③_（$info）;
?>
```

输出结果为：Array（[0]=>@mingrisoft[1]=>com[2]=>cn）

2．以下代码运行的结果为(　　　　　　)。

```
<?php
    $first="This course is very easy! ";
    $second=explode("",$first);
    $first=implode(",",$second);
    echo $first;
?>
```

3．PHP 同时支持两种风格的正则表达式，它们分别是（　　　　）和（　　　　）。

4．在 PHP 中，定义字符串需要使用特殊符号（　　　　）。

5．在 PHP 中，验证 IP 地址需要使用函数（　　　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

10.14　行 动 指 南

开始日期：______年____月____日　　　　　　　　　　　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。

续表

序号	内　　容	行 动 指 南	
1	自我测试栏目 分数（　　）	分数>75 分	优秀，有成为编程高手的潜质。
		分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）利用 mb_substr()函数实现中文字符串截取。	
		（2）利用 bin2hex()函数将二进制数据转换成十六进制数据。	
		（3）利用 strstr()函数检索字符串。	
		（4）利用正则表达式函数验证 IP 地址格式是否正确。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

10.15　成功可以复制——善于抓住时机的人徐少春

成就：2001 年，徐少春创建的金蝶国际软件集团有限公司在香港交易所顺利挂牌上市，并且他是金蝶软件最初版本的开发者。

创始公司：深圳爱普电脑技术有限公司、金蝶国际软件集团有限公司

财富巅峰：徐少春曾当选 2006 年度十大 IT 风云人物，此时的他早已经身价几十亿。

最高学历：硕士

出生日期：1963 年

超级语录：“在我的视线中，金蝶是不可挑战的。”

成功历程

徐少春年轻时家境贫寒，以至于高中时只能用咸菜萝卜、萝卜咸菜的无穷反复填满自己的肚子。但他决心要改变现状，甚至他在课桌的左右两角分别刻下“金钱在向你呼唤”、“美人在向你招手”，当然这也是那时大多数学生求学的动力。通过刻苦努力学习，徐少春最终考上了大学，并读取了研究生。

1988 年，徐少春被分配到地方税务局工作。时间到了 1990 年，徐少春放弃了人人羡慕的铁饭碗，来到改革开放的最前沿、创业的热土深圳，进了一家会计师事务所担任电脑部经理。

但就是电脑部经理这个职务，让徐少春更加清楚地知道，中国是多么需要财务管理软件。徐少春决定抓住这个时机，于是他辞掉电脑部经理的工作，并向岳父借了 5000 元买了一台 286 电脑，创办深圳爱普电脑技术有限公司，开始编写财务软件。那以后的两年间，徐少春带着独立开发的“爱普财务软件”，叩开一家又一家公司的门，亲自演示自己的软件产品。几个月后，他的用户增加到 30 多家，包括许多知名公司。凭借顽强的意志和坚定的信念，徐少春取得了最初的胜利。后来，徐少春先生与美籍华人赵西燕女士，以及深圳市蛇口工业区社会保险公司合资成立金蝶软件科技（深圳）有限公司，至此金蝶软件的名称才得以问世。



20 世纪 90 年代中期，当时 PC 机的作业系统还是 DOS 版的天下，Windows 平台的财务软件还少有人问津，甚至许多人都认为 Windows 时代还有 2～3 年才会到来。但徐少春又一次感受到 Windows 时代是大势所趋，不久降至，一定要抓住这个时机。于是在 1994 年，徐少春就开始带领金蝶研发 Windows 版财务软体，他和 20 多个开发人员春节都没有休息，到 1995 年初徐少春就拿出了金蝶财务软体 ForWindows 1.0 版。1996 年 4 月，金蝶 Windows 版财务软体被中国软体评测中心确认为中国首家优秀级 Windows 版财务软体。

20 世纪 90 年代末，在中国的南方，生产制造型企业不断兴起，企业的生产加工急需要软件来管理。在深圳的徐少春早就闻到企业 ERP 的气息，他又没错过这次时机，于是在 1999 年金蝶软件实现从财务软件向 ERP 软件的战略转移，这是一次革命性的转折点，为其以后成为中国软件产业的领导厂商奠定了基础。到了 2001 年，金蝶国际软件集团有限公司顺利在香港交易挂牌上市。

2007 年，随着电子商务的不断兴起，徐少春又开始实施金蝶软件“全面进军全程电子商务的计划”，于是推出了在线记账及商务管理平台——“友商网”。徐少春经过多年的拼搏与奋斗，时至今日，他所领导的金蝶国际软件集团有限公司已成为亚太地区领先的企业管理软件及电子商务应用解决方案供应商。

深度评价

从徐少春的成功历程中可以看出，他的成功之处在于很好地抓住每一次时机，这最终使金蝶在众多的财务软件公司中成为佼佼者。所以大家在学习和工作中，也要善于抓住各种时机，使我们个人的事业迈向更大的成功。

第 11 堂课

日期和时间的处理技术

（视频讲解：108 分钟）

在程序设计中，日期和时间是非常重要的，通过日期和时间可以记录数据库中数据的处理时间、计划的预计完成时间、系统当前时间、文件的操作时间，记录网站在某个时间段的访问量、系统管理员登录时间，网站遭到非法入侵的时间等。

学习摘要：

- PHP 的时间观念
- UNIX 时间戳
- 日期和时间的处理

11.1　PHP 的时间观念

在 PHP 语言中，日期、时间函数依赖于服务器的地区设置，而 PHP 默认设置的是标准的格林威治时间（即采用的是零时区）。如果没有对 PHP 的时区进行设置，并且用户的当地时间是北京时间，那么通过 PHP 的时间函数获取的时间就将比当地的北京时间少 8 个小时。因此，要获取本地当前的时间必须更改 PHP 语言中的时区设置。更改 PHP 语言中的时区设置有两种方法，即在 php.ini 文件中设置和通过 date_default_timezone_set 函数设置。

11.1.1　在 php.ini 文件中设置时区

在 php.ini 文件中设置时区，需要定位到[date]下的“;date.timezone=”选项，去掉前面的分号，并设置它的值为当地所在时区使用的时间。

例如，如果当地所在时区为东八区，就可以设置“date.timezone=”的值为 PRC（中华人民共和国）、Asia/Hong_Kong（香港）、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐）等，这些都是东八区的时间，如图 11.1 所示。

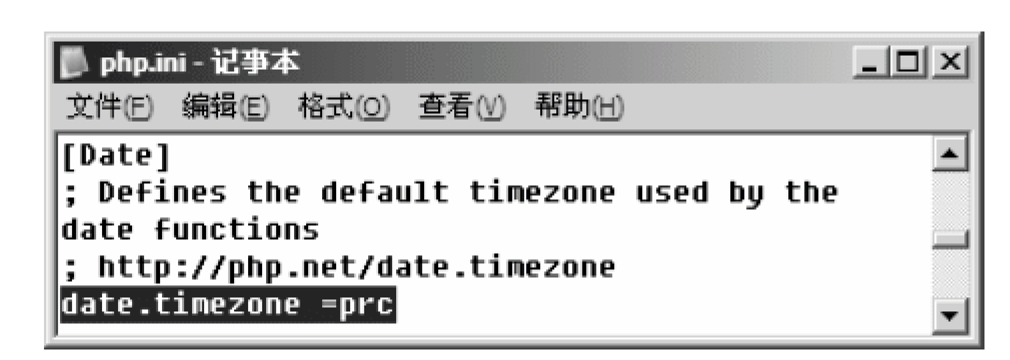


图 11.1　设置 PHP 的时区

设置完成后，保存文件，重新启动 Apache 服务器。

11.1.2　通过 date_default_timezone_set 函数设置时区

在应用程序中，在日期、时间函数之前使用 date_default_timezone_set()函数同样可以完成对时区的设置。date_default_timezone_set()函数的语法如下：

```
date_default_timezone_set(timezone);
```

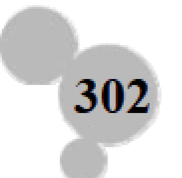
参数 timezone 为 PHP 可识别的时区名称，如果时区名称 PHP 无法识别，则系统采用 UTC 时区。

例如，设置北京时间可以使用的时区包括 PRC、Asia/Chongqing（重庆）、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐），这几个时区名称是等效的。

技巧：如果服务器使用的是零时区，则不能对 php.ini 文件直接进行修改，只能通过 date_default_timezone_set()函数对时区进行设置。

11.2　UNIX 时间戳

在日期、时间函数中，UNIX 时间戳的获取是非常重要的，有很多针对时间的操作都必须转换为时间戳



之后才能够完成。例如，比较时间大小、计算时间戳、实现倒计时的功能等。在 PHP 中提供了很多获取日期、时间戳的函数，如表 11.1 所示。

表 11.1　将日期、时间转换成 UNIX 时间戳的函数

函　数	说　明
gmmktime	获取 GMT（Greenwich Mean Time）日期的 UNIX 时间戳
microtime	返回当前 UNIX 时间戳和微秒数
mktime	获取一个日期的 UNIX 时间戳
strtotime	将任何英文文本的日期时间描述解析为 UNIX 时间戳
time	返回当前的 UNIX 时间戳

11.2.1　获取任意日期、时间的时间戳

mktime()函数将一个时间转换成 UNIX 时间戳。其语法如下：

```
int mktime(int hour, int minute, int second, int month, int day, int year, int [is_dst] )
```

mktime()函数根据给出的参数返回 UNIX 时间戳。其参数可以从右向左省略，任何省略的参数都会被设置成本地日期、时间的当前值（即不设置任何参数，mktime()函数获取的是本地当前日期和时间）。mktime()函数的参数说明如表 11.2 所示。

表 11.2　mktime()函数的参数说明

参　数	说　明
hour	小时数
minute	分钟数
second	秒数（一分钟之内）
month	月份数
day	天数
year	年份数，可以是两位或 4 位数字，0～69 对应于 2000～2069，70～100 对应于 1970～2000
is_dst	参数 is_dst 在夏令时可以被设为 1，如果不是则设为 0；如果不确定是否为夏令时则设为-1（默认值）

说明：有效的时间戳范围是格林威治时间 1901 年 12 月 13 日 20:45:54 到 2038 年 1 月 19 日 03:14:07。（此范围符合 32 位有符号整数的最小值和最大值）。在 Windows 系统中此范围限制为从 1970 年 1 月 1 日到 2038 年 1 月 19 日。

例如，应用 mktime()函数获取当前时间的时间戳。代码如下：

```
<?php
echo mktime();                                   //当前时间戳
?>
```

运行结果为：1287628331

11.2.2　获取当前时间戳

11.2.1 小节讲解的 mktime()函数在不设置任何参数的情况下可以获取当前时间的时间戳，但是 PHP 也

提供了专门的获取当前时间时间戳的函数，那就是 time()函数。

time()函数获取当前的 UNIX 时间戳，返回值为从 UNIX 纪元（格林威治时间 1970 年 1 月 1 日 00:00:00）到当前时间的秒数。语法如下：

```
int time ( void )
```

time()函数没有参数，返回值为 UNIX 时间戳的整数值。

例如，应用 time()函数获取当前时间的时间戳。代码如下：

```
<?php
echo time();                                        //当前时间戳
?>
```

运行结果为：1287628348

11.2.3 日期、时间转换为 UNIX 时间戳

strtotime()函数将任何英文文本的日期时间描述解析为 UNIX 时间戳。其语法如下：

```
int strtotime ( string time [, int now] )
```

strtotime()函数接受一个包含英语日期格式的字符串并尝试将其解析为 UNIX 时间戳（自 January 1 1970 00:00:00 GMT 起的秒数），其值相对于 now 参数给出的时间，如果没有提供此参数则用系统当前时间。

如果参数 time 的格式是绝对时间，则 now 参数不起作用；如果参数 time 的格式是相对时间，那么其对应的时间由参数 now 来提供。

如果解析成功，则返回时间戳，否则返回 FALSE。在 PHP 5.1.0 之前本函数在失败时返回−1。

例如，应用 strtotime()函数将当前时间和指定日期转换为时间戳。代码如下：

```
<?php
echo strtotime ("now"), "\n";                                             //当前时间的时间戳
echo "输出时间:".date("Y-m-d H:i:s",strtotime ("now")),"<br>";            //输出当前时间
echo strtotime ("20 October 2010"), "\n";                                 //输出指定日期的时间戳
echo "输出时间:".date("Y-m-d H:i:s",strtotime ("20 October 2010")),"<br>"; //输出指定日期的时间
?>
```

运行结果为：

1287628521 输出时间:2010-10-21 02:35:21

1287532800 输出时间:2010-10-20 00:00:00

提示： 如果给定的年份是两位数字的格式，则其值 0～69 表示 2000—2069，70～100 表示 1970—2000。有效的时间戳通常从 Fri, 13 Dec 1901 20:45:54 GMT 到 Tue, 19 Jan 2038 03:14:07 GMT（对应于 32 位有符号整数的最小值和最大值）。不是所有的平台都支持负的时间戳，那么日记范围就被限制为不能早于 UNIX 纪元。这意味着在 1970 年 1 月 1 日之前的日期将不能用在 Windows、一些 Linux 版本，以及几个其他的操作系统中。不过 PHP 5.1.0 及更新的版本克服了此限制。

11.3 日期和时间的处理

日期和时间的处理可以分为格式化日期和时间、获取日期和时间信息、获取本地化的日期和时间和检验日期和时间的有效性等。日期和时间处理函数如表 11.3 所示。



表 11.3　日期和时间处理函数

函　　数	说　　明
checkdate	验证日期的有效性
date	格式化一个本地时间/日期
getdate	获取日期/时间信息
gettimeofday	获取当前时间
gmdate	格式化一个 GMT（格林威治标准时间）/UTC 日期/时间
gmstrftime	根据区域设置格式化 GMT/UTC 时间/日期
localtime	获取本地时间
strftime	根据区域设置格式化本地时间/日期

11.3.1　格式化日期和时间

date()函数对本地日期和时间进行格式化。其语法如下：

```
date(string format,int timestamp)
```

参数 format 指定日期和时间输出的格式。有关参数 format 指定的格式如表 11.4 所示。参数 timestamp 为可选参数，指定时间戳，如果没有指定时间戳则使用本地时间戳 time()。

表 11.4　date()函数中参数 format 的格式选项

参　　数	说　　明
A	小写的上午和下午值，返回值为 am 或 pm
A	大写的上午和下午值，返回值为 AM 或 PM
B	Swatch Internet 标准时间，返回值为 000～999
d	月份中的第几天，有前导零的两位数字，返回值为 01～31
D	星期中的第几天，文本格式，3 个字母，返回值为 Mon～Sun
F	月份，完整的文本格式，返回值为 January～December
G	小时，12 小时格式，没有前导零，返回值为 1～12
H	小时，24 小时格式，没有前导零，返回值为 0～23
I	有前导零的分钟数，返回值为 00～59
I	判断是否为夏令时，返回值如果是夏令时为 1，否则为 0
J	月份中的第几天，没有前导零，返回值为 1～31
l	星期数，完整的文本格式，返回值为 Sunday～Saturday
L	判断是否为闰年，返回值如果是闰年为 1，否则为 0
m	数字表示的月份，有前导零，返回值为 01～12
M	3 个字母缩写表示的月份，返回值为 Jan～Dec
n	数字表示的月份，没有前导零，返回值为 1～12
o	与格林威治时间相差的小时数，如 0200
r	RFC 822 格式的日期，如 Thu，21 Dec 2000 16:01:07 +0200
s	秒数，有前导零，返回值为 00～59
S	每月天数后面的英文后缀，两个字符，如 st、nd、rd 或者 th。可以和 j 一起使用
t	指定月份所应有的天数

续表

参　　数	说　　明
T	本机所在的时区
U	从 UNIX 纪元（January 1 1970 00:00:00 GMT）开始至今的秒数
w	星期中的第几天，数字表示，返回值为 0～6
W	ISO-8601 格式年份中的第几周，每周从星期一开始
y	两位数字表示的年份，返回值如 88 或 08
Y	4 位数字完整表示的年份，返回值如 1998、2008
z	年份中的第几天，返回值为 0～366
Z	时差偏移量的秒数。UTC 西边的时区偏移量总是负的，UTC 东边的时区偏移量总是正的，返回值为 -43200～43200

例如，应用 date()函数设置不同的 format 值，输出不同格式的时间。代码如下：

```
<?php
echo "单个变量：".date("m 月");                                    //输出单个日期
echo "<p>";
echo "组合变量：".date("Y-m-d");                                   //输出组合参数
echo "<p>";
echo "详细的日期及时间：".date("Y-m-d H:i:s");                      //输出详细的日期和时间参数
echo "<p>";
echo "中文格式日期及时间：".date("Y 年 m 月 d 日  H 时 i 分 s 秒");   //输出中文格式时间
?>
```

运行结果如图 11.2 所示。



图 11.2　date()函数输出不同格式的当前时间

提示：在运行本堂课的实例时，也许有的读者得到的时间和系统时间并不相等，这不是程序的问题。因为在 PHP 语言中默认设置的是标准的格林威治时间，而不是北京时间。如果出现了时间不符的情况，那么请参考本堂课 11.1 节的内容。

11.3.2　获取日期和时间信息

getdate()函数获取日期和时间指定部分的相关信息。其语法如下：

```
array getdate(int timestamp)
```

getdate 函数返回数组形式的日期、时间信息，如果没有时间戳，则以当前时间为准。getdate 函数返回的关联数组元素的说明如表 11.5 所示。



表 11.5　getdate()函数返回的关联数组元素的说明

键　名	说　明	返　回　值
seconds	秒	返回值为 0～59
minutes	分钟	返回值为 0～59
hours	小时	返回值为 0～23
mday	月份中第几天	返回值为 1～31
wday	星期中第几天	返回值为 0（表示星期日）～6（表示星期六）
mon	月份数字	返回值为 1～12
year	4 位数字表示的完整年份	返回的值如 2010 或 2011
yday	一年中第几天	返回值为 0～365
weekday	星期几的完整文本表示	返回值为 Sunday～Saturday
month	月份的完整文本表示	返回值为 January～December
0	自从 UNIX 纪元开始至今的秒数，和 time()的返回值以及用于 date()的值类似	系统相关，典型值为从-2147483648～2147483647

getdate()函数比较适合获取当前日期是一年、月份或者星期中的第几天。虽然它也可以获取当前的日期，但是由于获取的是返回值数组，所以更适合获取时间中某个特定的值。

例如，通过 getdate()函数获取当前日期，以及当前日期是一年中的第几天和当月的第几天。代码如下：

```
<?php
$arr = getdate();                                        //使用 getdate()函数将当前信息保存
$month=date("m 月");                                     //输出单个日期
$year=date("Y 年");
echo "当前日期：".$arr[year]."-".$arr[mon]."-".$arr[mday]." ";   //返回当前的日期信息
echo "<P>";
echo "今天是".$year."中的第".$arr[yday]."天";              //输出今天是一年中的第几天
echo "<p>";
echo "今天是".$year.$month."的第".$arr[mday]."天";         //输出今天是本月中的第几天
?>
```

运行结果如图 11.3 所示。

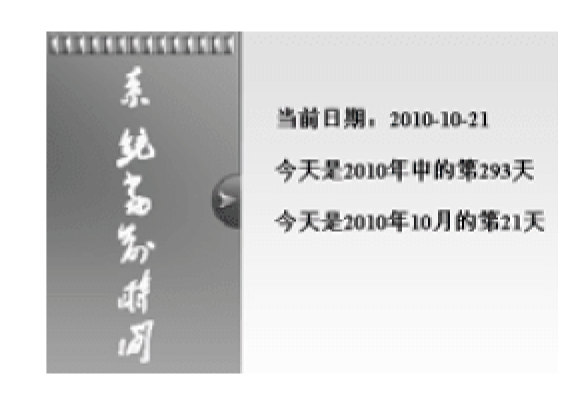


图 11.3　getdate()函数获取当前时间信息

11.3.3　检验日期和时间的有效性

一年 12 个月、一个月 31 天（或 30 天，2 月 28 天，闰年为 29 天），一星期 7 天……这些都是基本常识。但计算机并不能自己分辨数据的对与错，只是依靠开发者提供的功能去执行或检查。在 PHP 中通过 checkdate()函数检验日期和时间的有效性。其语法如下：

```
bool checkdate(int month,int day,int year)
```

其中，参数 month 的有效值为 1～12；参数 day 的有效值为当月的最大天数，例如，1 月为 31 天，2 月

为 29 天（闰年）；参数 year 的有效值为 1～32767。如果验证的日期有效则返回 TRUE，否则返回 FALSE。

例如，验证 2010 年 2 月份到底是 28 天，还是 29 天。代码如下：

```
<?php
    $year = 2010;                              //2010 年
    $month = 2;                                //2 月份
    $one_day = 28;                             //28 天
    if(checkdate($month,$one_day,$year)){
        echo "2010 年 2 月份是 28 天";
    }else{
        echo "2010 年 2 月份是 29 天";
    }
?>
```

11.4 照猫画虎——基本功训练

11.4.1 基本功训练 1——获取指定时间的 UNIX 时间戳

视频讲解：光盘\mr\11\lx\获取指定时间的 UNIX 时间戳.exe

实例位置：光盘\mr\11\zmhh\01

获取指定时间的时间戳需要使用 strtotime()函数，此函数接受任何一个英文文本的日期时间并将其转换为时间戳，本实例以 10 September 2010 为例获取此时刻的时间戳，运行结果如图 11.4 所示。

10 September 2010的时间戳为 1284076800

图 11.4 获取指定时间的时间戳

实现原理非常简单，在 strtotime()函数中添加指定时间参数即可。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                                   //设置编码集
    $str="<font style='color:red;font-size:20px'>".strtotime("10 September 2010")."</font>";   //指定获取时间戳
    echo "10 September 2010 的时间戳为 ";
    echo $str;
?>
```

照猫画虎：利用 date()函数获取当前时间的时间戳。（20 分）（实例位置：光盘\mr\11\zmhh\01_zmhh）

11.4.2 基本功训练 2——获取系统当前时间的时间戳

视频讲解：光盘\mr\11\lx\获取系统当前时间的时间戳.exe

实例位置：光盘\mr\11\zmhh\02

通过本实例检验读者对日期、时间函数的掌握情况，应用 date()和 time()函数获取系统当前时间和时间戳。本实例的运行结果如图 11.5 所示。



图 11.5 获取当前时间的时间戳

应用 date()函数获取当前时间，应用 time()函数获取当前时间戳。代码如下：

```
<?php
    echo date("Y 年 m 月 d 日  H 时 i 分 s 秒");        //获取当前时间
```



```
    echo "<br>";                              //换行
    echo time();                              //获取当前时间戳
?>
```

照猫画虎：利用 strtotime()函数获取当前时间的时间戳。（20 分）（实例位置：光盘\mr\11\zmhh\02_zmhh）

11.4.3　基本功训练 3——输出系统当前时间

视频讲解：光盘\mr\11\lx\输出系统当前时间.exe

实例位置：光盘\mr\11\zmhh\03

系统的当前时间是服务器系统的当前时间，此时间是以格林威治时间为基准的，也就是说获取到的时间与中国大陆的时间会相差 8 个小时。本实例的运行结果如图 11.6 所示。

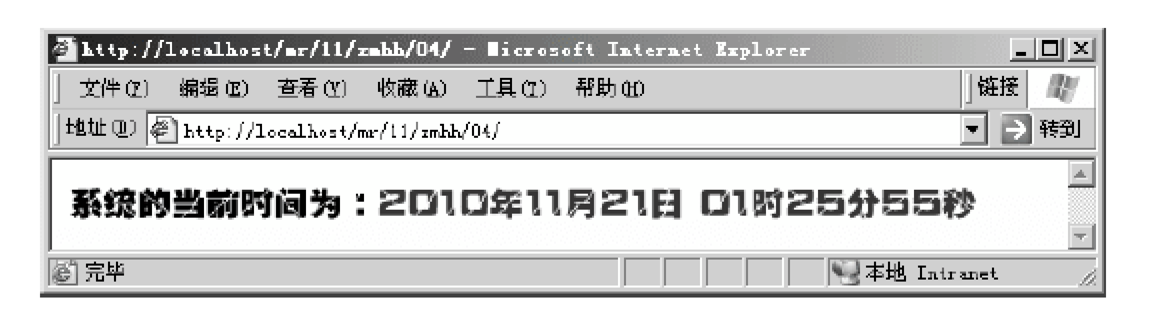


图 11.6　输出系统当前时间

实现本实例应用到的函数是 date()函数，应用此函数可直接获取当前系统的时间。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");            //设置编码集
    echo "<font style='color:black;font-family:文鼎霹雳体;font-size:20px'>系统的当前时间为：</font>";
    echo "<font style='color:red;font-family:文鼎霹雳体;font-size:20px'>".date("Y 年 m 月 d 日   H 时 i 分 s 秒").
"</font>";                                                                  //输出当前系统时间
?>
```

照猫画虎：输出当前系统日期下的日出时间。（20 分）（实例位置：光盘\mr\11\zmhh\03_zmhh）

11.4.4　基本功训练 4——倒计时

视频讲解：光盘\mr\11\lx\倒计时.exe

实例位置：光盘\mr\11\zmhh\04

倒计时是大家在生活中经常会用到的一个功能，例如，2010 年上海世博会的倒计时，2011 年春节的倒计时等。下面应用 PHP 的日期、时间函数为 2011 年元旦设计一个倒计时程序。本实例的运行结果如图 11.7 所示。

距离2011年元旦还有72 天!!!

图 11.7　倒计时

其原理非常简单，就是用一个固定的时间减去当前的时间，所得到的就是剩余时间。要完成时间的加减操作，同比较大小类似，都要将时间转换成时间戳，然后才能计算，最后再将时间戳转换成日期输出。代码如下：

```
<?php
    $time1 = strtotime(date( "Y-m-d"));                    //当前的系统时间
    $time2 = strtotime("2011-1-1");                        //2011 年元旦
    $sub2 = ceil(($time2 - $time1) / 86400);               //(60 秒*60 分*24 小时)秒/天
     echo "距离 2011 年元旦还有<font color=red>$sub2 </font>天!!!";
?>
```

照猫画虎：获取 2010 年元旦是星期几。（20 分）（实例位置：光盘\mr\11\zmhh\04_zmhh）

11.4.5 基本功训练 5——获取本地时间

视频讲解：光盘\mr\11\lx\获取本地时间.exe

实例位置：光盘\mr\11\zmhh\05

前面的实例中已经提到过格林威治时间与中国本地时间相差 8 个小时，所以如果想获取当前的本地时间就存在着两种方法：第一种方法，系统的当前时间加上 8 个小时即可。但是这种方法对于程序员来说相当幼稚。第二种方法，利用 date_default_ timezone_set()函数实现时区的设置，然后获取当前时间。本实例采用第二种方法实现，运行结果如图 11.8 所示。

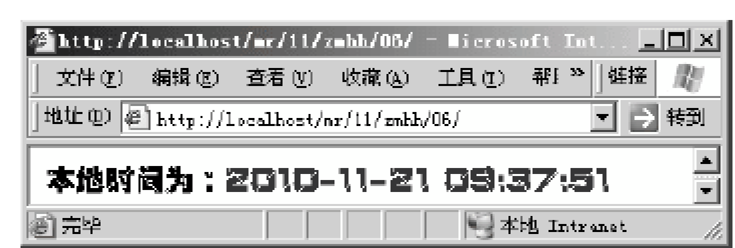


图 11.8 获取本地时间

本实例实现过程非常简单，需要使用 date_default_timezone_set()函数设置时区后利用 date()函数获取当前日期和时间。代码如下：

```
<?php
     header("Content-Type:text/html;charset=utf-8");                    //设置编码集
     date_default_timezone_set("Asia/ShangHai");                         //设置当前时区
      //获取本地时间
     echo "<font style='color:black;font-family:文鼎霹雳体;font-size:18'>本地时间为：</font>";
     $str="<font style='color:red;font-family:文鼎霹雳体;font-size:18'>".date("Y-m-d H:i:s")."</font>";
     echo $str;
?>
```

照猫画虎：获取本地时间当前时刻的微秒数。（20 分）（实例位置：光盘\mr\11\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

11.5 情景应用——拓展与实践

11.5.1 情景应用 1——比较两个时间的大小

视频讲解：光盘\mr\11\lx\比较两个时间的大小.exe

实例位置：光盘\mr\11\qjyy\01

在 PHP 语言中，要完成对两个时间大小的比较，必须先将时间转换为时间戳，然后才可以进行比较。而将时间转换为时间戳可以通过 strtotime()函数完成，运行结果如图 11.9 所示。



图 11.9 比较两个时间戳大小

首先，定义两个固定的时间，然后通过 strtotime()函数将固定时间转换成时间戳，最后对两个时间戳的值进行比较。代码如下：

```
<?php
      $time = "2010 年 10 月 10 日  10 时 10 分 10 秒";          //设置固定时间
      $times = "2011-11-11 11:11:11";                            //设置固定时间
      if(strtotime($time) - strtotime($times) < 0){              //对两个时间进行运算
           echo "时间：".$time."<p>  早于 ".$times ;             //time－times<0 说明 time 的时间在前
```

```
    }else{
        echo "时间：".$times."<p> 早于 ".$time;              //否则，说明 times 的时间在前
    }
?>
```

DIY：看看从格林威治时间到现在一共经历了多少天。（20 分）（实例位置：光盘\mr\11\qjyy\01_diy）

11.5.2　情景应用 2——网页闹钟

视频讲解：光盘\mr\11\lx\网页闹钟.exe

实例位置：光盘\mr\11\qjyy\02

闹钟是生活中经常使用的一个小工具，它会在指定时间叫你起床。在 Web 程序中也可以应用这个原理，开发一个“网页闹钟”，提示用户在指定的时间或者时间段内要做什么工作。本实例是当系统的当前时间运行到 9 月 18 日，给出提示信息“勿忘国耻！”，运行结果如图 11.10 所示。



图 11.10　提示信息

所谓网页闹钟也就是一个日志提醒功能，通过判断系统中当前的时间与指定的某个时间或者时间段是否相同，如果相同系统就给出一个对应的提示信息，提示用户该做什么（如每天工作记录的上传、每月经验技巧的提交或者员工生日的提醒等），都可以通过日志的形式进行提醒。在本实例中，通过当前时间戳与 9 月 18 日时间戳进行比较，如果相同则给出提示信息。关键代码如下：

```
<?php
    $time1 = strtotime(date("Y-m-d"));                     //当前的系统时间，获取月和天的时间戳
    $time2 = strtotime(date("Y")."-09-18");                //设置时间 3 月 15 日的时间戳
    if($time1==$time2){                                    //判断两个时间戳是否相同
        echo "<script>alert('勿忘国耻！');window.location.href='index.php';</script>";    //给出提示信息
    }else{
        echo "今天不是一个特殊的日子！";
    }
?>
```

DIY：网页闹钟程序中输出当前是星期几。（20 分）（实例位置：光盘\mr\11\qjyy\02_diy）

11.5.3　情景应用 3——计算程序的运行时间

视频讲解：光盘\mr\11\lx\计算程序的运行时间.exe

实例位置：光盘\mr\11\qjyy\03

在百度中，当大家执行一个查询操作时，当获取到查询结果后，页面中就会出现一行文字，提示用户根据关键字搜索到多少个结果，以及搜索所用的时间。这个时间就是程序在执行这个搜索时所用的时间，那么它是如何实现的呢？下面模仿它来做一个计算程序运行时间的小程序，运行结果如图 11.11 所示。

距离2011年元旦还有72 天!!!

图 11.11　计算程序运行时间

其原理是：首先，在执行查询操作之前定义一个时间，将其精确到微秒，然后执行查询操作，接着，在查询功能执行完毕后，再次获取一个时间，同样精确到微秒，最后，应用后获取的时间减去最初获取的时间，就是本次查询所用的时间。关键代码如下：

```
<?php
/*	声明 run_time()函数
通过 microtime()获取当前时间的微秒数和时间戳
应用 explode()函数将 microtime()函数返回的字符串进行分隔，返回一个数组
包括两个元素，一个元素是当前时间的微秒数，另一个是当前时间的时间戳
应用 list()函数将 explode()函数返回的数组值赋给指定的变量
最后，获取两个变量值的和
*/
function run_time(){
list($msec, $sec) = explode(" ", microtime());                //使用 explode 函数返回两个变量
return ((float)$msec + (float)$sec);                          //返回两个变量的和
}
$start_time = run_time();                                     //第一次运行 run_time()函数
/*	运行 PHP 代码段	*/
$end_time = run_time();                                       //再次运行 run_time()函数
?>
```

DIY：根据上面的原理，计算执行 100 次循环所用的时间。（20 分）（实例位置：光盘\mr\11\qjyy\03_diy）

11.5.4 情景应用 4——输出中文格式的日期和时间

视频讲解：光盘\mr\11\lx\输出中文格式的日期和时间.exe

实例位置：光盘\mr\11\qjyy\04

输出中文格式的时间其实在前面的操作中已经提到过。虽然本实例的知识点不是很大，但是值得再次提及到。本实例采用 date()函数实现中文格式的日期和时间的输出，运行结果如图 11.12 所示。



图 11.12 输出中文格式的日期和时间

本实例是采用 date()函数，定义传递到此函数的参数为“Y 年 m 月 d 日 H 时 i 分 m 秒”。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");          //设置编码集
	date_default_timezone_set("Asia/Shanghai");              //设置时区
	echo date("Y 年 m 月 d 日  H 时 i 分 m 秒");                //输出信息
?>
```

DIY：输出当前日期是一年的第几天。（20 分）（实例位置：光盘\mr\11\qjyy\04_diy）

11.5.5 情景应用 5——检验日期和时间的有效性

视频讲解：光盘\mr\11\lx\检验日期和时间的有效性.exe

实例位置：光盘\mr\11\qjyy\05

一年 12 个月、一个月 31 天（或 30 天，2 月 28 天，闰年为 29 天），一星期 7 天……这些都是基本常识。但计算机并不能自己分辨数据的对与错，只是依靠开发者提供的功能去执行或检查。在 PHP 中通过 checkdate()函数检验日期和时间的有效性。本实例的运行结果如图 11.13 所示。



图 11.13　检验日期和时间的有效性

本实例是采用 checkdate()函数传递 3 个参数判断某年某月是否存在某日的操作。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");
	$year = 2010;								//2010 年
	$month = 2;								//2 月份
	$one_day = 28;								//28 天
	if(checkdate($month,$one_day,$year)){
		echo "<font style='color:black;font-family:文鼎霹雳体;font-size:25'>2010 年 2 月份是 28 天</font>";
	}else{
		echo "<font style='color:black;font-family:文鼎霹雳体;font-size:18'>2010 年 2 月份是 29 天</font>";
	}
?>
```

DIY：让本程序休眠 10 秒后再检测日期和时间的有效性。（20 分）（实例位置：光盘\mr\11\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

11.6　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．下面（　　）函数可以实现当地时间的设置。

　A．date　　B．time　　C．date_default_timezone_set

2．获取当前时间戳需要使用的函数是（　　）。

　A．date　　B．time

　C．date_default_timezone_set　　D．substr

3．检验日期和时间的有效性需要使用函数（　　）。

　A．date　　B．time　　C．checkdate

4．获取时间戳的微秒数需要使用函数（　　）。

　A．time　　B．date　　C．microtime

5．看下面一行代码：

```
echo date("Y-m-d")
```

程序运行后，输出日期的格式为（　　）。

　A．X 年 X 月 X 日　　B．X - X - X　　C．X - X - X　X: X:X

二、填空题（每题 10 分，5 道题）

1．获取 2010 年 1 月 1 日的时间戳代码为（   ）。
2．将系统时间设置为当前本地时间的代码为（   ）。
3．利用 mktime()函数获取当前时间的代码为（   ）。
4．计算 2010 年 12 月 10 日是星期几的代码为（   ）。
5．程序运行后，下面代码显示的内容为（   ）。

```
<?php
	$time1 = strtotime(date("Y-m-d"));
	$time2 = strtotime("2008-09-18");
	if($time1==$time2){
		echo "<script>alert('勿忘国耻！');window.location.href='index.php';</script>";	//给出提示信息
	}else{
		echo "今天不是一个特殊的日子！";
	}
?>
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

11.7 行动指南

开始日期：______年____月____日　　　　　　　　　　结束日期：______年____月____日

序号	内　容	行动指南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）制作一个万年历。	
		（2）制作一个日期温馨小提示程序。	
		（3）制作一个小程序，判断当前是一年的第多少天。	
		（4）制作一个小程序，提醒今天是星期几。	

续表

序号	内　容	行 动 指 南
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

11.8　成功可以复制——C#语言之父安德斯·海尔斯伯格

成就：开发了享誉世界的 Turbo Pascal、Delphi、C#等编程语言，安德斯曾被《Dr.Dobb's》杂志授予“Prestigious Excellence in Programming Award”奖。

创始公司：丹麦的 Poly Data 公司、美国的 Borland 公司的创始人之一

财富巅峰：在编程语言设计上的巨大成功，成就了他亿万富翁的身价。

最高学历：本科

出生日期：1960 年 11 月

超级语录：“每两年，CPU 的速度要加倍，内存容量是原来的四倍，但是我们的大脑还是原来的大小，很显然，将需要更多的程序员来补充这些。”

成功历程

安德斯·海尔斯伯格（Anders Hejlsberg）曾在丹麦技术大学学习工程，但没有毕业。他自学计算机编程，并在丹麦创建了 Poly Data 公司，凭借他的聪明才智和富有挑战的性格，独立开发了 Pascal 语言的编译器核心。

20 世纪 80 年代，安德斯去了美国，他凭借掌握的 Pascal 语言的编译器核心技术和著名数学家 Philippe Kahn 共同创建了世界著名的第三大软件公司——Borland 公司。

安德斯首先在 Borland 主持开发了享誉盛名的 Turbo Pascal 语言，Turbo Pascal 语言的设计成功使得他的名声享誉全球 IT 界。但安德斯并未因此而沾沾自喜，止步不前，而是接着挑战自己目前的成功。接下来，安德斯带领他的研发小组继续升级 Turbo Pascal 语言，最终将 Turbo Pascal 变成一种面向对象的、拥有可视化环境和卓越的数据库访问特性的应用程序开发语言——Delphi，发布后的 Delphi 语言立刻在全球 IT 界获得了开发工具史上奇迹之称的美誉。

20 世纪 90 年代，Java 语言出现了，安德斯在 Borland 一直感觉郁郁不得志，这时慧眼识才的比尔•盖茨，三顾茅庐，邀请安德斯加入微软。最开始，微软的重金开价并未打动安德斯，因为凭借他在 Borland 的多年工作，并作为公司的创始人之一，他应该拥有大量的财富，早已是亿万富翁级的身价。实际上，在安德斯的内心深处更需要一个宽松的工作环境和编程领域内更大的挑战，但他的这些想法，当时的 Borland 公司是无法给予他的。当比尔•盖茨知道他的想法后，马上给他一个非常大的挑战，许诺给他一个宽松的工作环境，让其领导 Visual J++语言设计小组，这样他欣然加入了。在安德斯的带领下，Visual J++语言的开发非常成功，很快 Visual J++将 Java 开发人员拉拢到它的周围。于是在 1999 年安德斯被授予“distinguished engineer”（卓越工程师），在微软仅有 16 人获得这样的荣誉。



后来 SUN 公司以微软破坏了 Java 的跨平台性为由，终止了对微软的 Java 授权，这样促使微软决心自己研发能够面向网络的、完全面向对象的编程语言——C#。微软把这个在编程领域具有革命意义的重任交给了安德斯，安德斯又一次非常高兴地接受了挑战。而后，安德斯带领他的 C#研发团队，经过几年时间的埋头苦干，终于使 C#成为微软反击 Java 语言的最有力武器。

深度评价

从安德斯编写 Pascal 语言的编译器核心开始，到他后来设计 Turbo Pascal 语言、Delphi 语言、VJ++语言、C#语言可以看到，他的每一步都获得巨大成功，但他从未止步不前，而是把每一次的成功作为迎接更大挑战的基石。安德斯的这种富于挑战，不断超越自我的精神，非常值得我们大家学习，学习编程也一样，只有不断地向上攀登，才会有更大的收获和成功。

第12堂课

图形图像处理技术

（视频讲解：125分钟）

由于有GD库的强大支持，PHP的图像处理功能可以说是PHP的一个强项，便捷易用、功能强大。另外，PHP图形化类库——Jpgraph也是一款非常好用和强大的图形处理工具，可用来绘制各种统计图和曲线图，也可以自定义设置颜色和字体等元素。

图像处理技术中的经典应用是绘制饼形图、柱形图和折线图，这是对数据进行图形化分析的最佳方法。本堂课将分别对GD2函数及Jpgraph类库进行详细讲解。

学习摘要：

- 了解GD2函数库
- 设置GD2函数库
- 学习常用图像处理技术
- 运用Jpgraph类库绘制图像

12.1　了解 GD2 函数库

PHP 目前在 Web 开发领域已经被广泛的应用，互联网上已经有近半数的站点采用 PHP 作为核心语言。PHP 不仅可以生成 HTML 页面，而且可以创建和操作二进制形式的数据，如图像、文件等，其中使用 PHP 操作图形可以通过 GD2 函数库来实现，使用 GD2 函数库可以在页面中绘制各种图形图像，以及统计图，如果与 Ajax 技术相结合，还可以制作出各种强大的动态图表。

GD2 函数库是一个开放的、动态创建图像的、源代码公开的函数库，可以从官方网站“http://www.boutell.com/gd”下载最新版本的 GD2 库。目前，GD2 库支持 GIF、PNG、JPEG、WBMP 和 XBM 等多种图像格式。

12.2　设置 GD2 函数库

PHP 5 中 GD2 函数库已经作为扩展被默认安装，但目前有些版本中，还需要通过对 php.ini 文件进行设置来激活 GD2 函数库。使用文本编辑工具，如记事本等打开 php.ini 文件，将该文件中的“;extension=php_gd2.dll”选项前的分号“;”删除，如图 12.1 所示。保存修改后的文件，并重新启动 Apache 服务器，即可激活 GD2 函数库。

在成功加载 GD2 函数库后，可以通过 phpinfo()函数来获取 GD2 函数库的安装信息，验证 GD 库是否安装成功。在 Apache 的默认站点目录中编写 phpinfo.php 文件，并在该文件中编写如下代码：

```
<?php
    phpinfo();       //输出 PHP 配置信息
?>
```

在 IE 浏览器的地址栏中输入“http://127.0.0.1/phpinfo.php”，按 Enter 键后，如果在打开的页面中检索到如图 12.2 所示的 GD 库安装信息，说明 GD 库安装成功，这样开发人员就可以在程序中使用 GD2 函数库编写图形图像。



图 12.1　激活 GD2 函数库

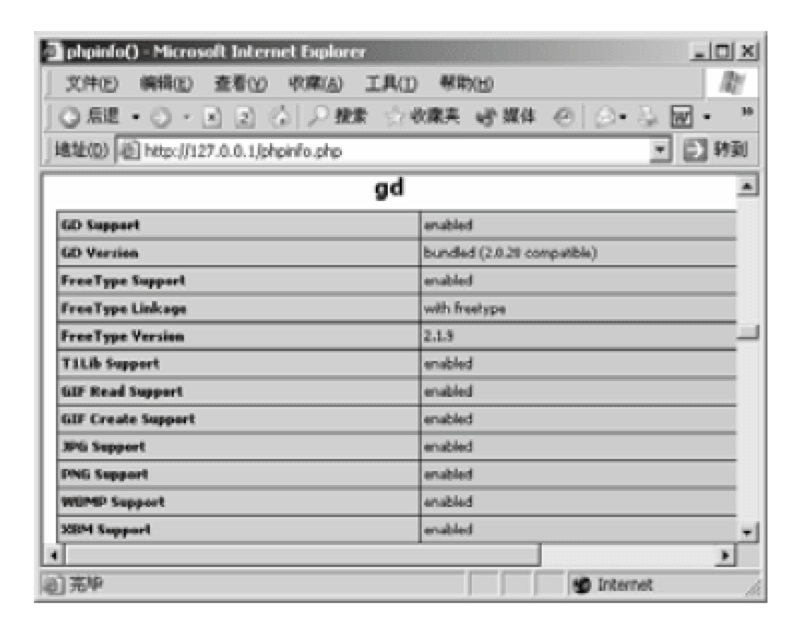
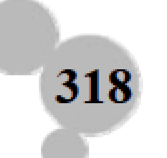


图 12.2　GD2 函数库的安装信息

12.3　学习常用图像处理技术

PHP 中的 GD2 函数库用于创建或者处理图片，通过它可以生成统计图表、动态图形、缩略图和图形验



证码等。在 PHP 中，对图像的操作可以分为以下 4 个步骤：

☑　创建画布。

☑　在画布上绘制图形。

☑　保存并输出结果图像。

☑　销毁图像资源。

12.3.1　创建画布

GD2 函数库在图形图像绘制方面功能非常强大，利用它开发人员既可以在已有图片的基础上进行绘制，也可以在没有任何素材的基础上绘制，在后者情况下首先要创建画布，之后所有操作都将依据所创建的画布进行。在 GD2 函数库中创建画布应用 imagecreate()函数，语法如下：

```
resource imagecreate ( int x_size, int y_size )
```

该函数用于返回一个图像标识符，参数 x_size、y_size 为图像的尺寸，单位为像素（pixel）。

例 12.01　通过 imagecreate()函数创建一个宽 200 像素，高 100 像素的画布，并且设置画布背景颜色 RGB 值为 211、126、29，最后输出一个 jpg 格式的图像，具体代码如下。**（实例位置：光盘\mr\12\sl\12.01）**

```
<?php
    header("Content-type:text/html;charset=utf-8");         //设置页面的编码风格
    header("Content-type:image/jpg");                       //告知浏览器输出的是图片
    $image=imagecreate(400,100);                            //设置画布的大小
    $bgcolor=imagecolorallocate($image,200,60,60);          //设置画布的背景颜色
    imagejpeg($image);                                      //输出图像
    imagedestroy($image);                                   //销毁图像
?>
```

在上面的代码中，应用 imagecreate()函数创建一个基于普通调色板的画布，通常支持 256 色。其中通过 imagecolorallocate()函数设置画布的背景颜色；通过 imagegif()函数输出图像；通过 imagedestroy()函数销毁图像资源。运行结果如图 12.3 所示。



图 12.3　创建画布

12.3.2　颜色处理

应用 GD2 函数库绘制图形需要为图形中的背景、边框和文字等元素指定颜色，GD2 中使用 imagecolor allocate()函数设置颜色，语法如下：

```
int imagecolorallocate ( resource image, int red, int green, int blue)
```

其中，image 参数是 imagecreatetruecolor()函数的返回值。red、green 和 blue 分别是所需要的颜色的红、绿、蓝成分，这些参数是 0～255 的整数或者十六进制的 0x00～0xFF。

imagecolorallocate()函数返回一个标识符，代表由给定的 RGB 成分组成的颜色。

提示：如果是第一次调用 imagecolorallcate()函数，那么它将完成背景颜色的填充。

例 12.02 通过 imagecreate()函数创建一个宽 685 像素、高 180 像素的画布，通过 imagecolorallocate()函数为画布设置背景颜色以及图像的颜色，并且输出创建的图像。**（实例位置：光盘\mr\12\sl\12.02）**

首先，通过 imagecreate()函数创建一个宽 685 像素、高 180 像素的画布。然后，通过 imagecolorallocate()函数为画布设置背景颜色以及图像的颜色。接着通过 imageline()函数绘制一条白色的直线。最后，完成图像的输出和资源的销毁。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");          //设置页面的编码风格
    header("Content-Type:image/jpeg");                       //告知浏览器输出的是一个图片
    $image=imagecreate(685,180);                             //设置画片大小
    $bgcolor=imagecolorallocate($image,200,60,120);          //设置图片的背景颜色
    $write=imagecolorallocate($image,200,200,250);           //设置线条的颜色
    imageline($image,20,20,650,160,$write);                  //画一条线
    imagejpeg($image);                                       //输出图像
    imagedestroy($image);                                    //销毁图片
?>
```

运行结果如图 12.4 所示。

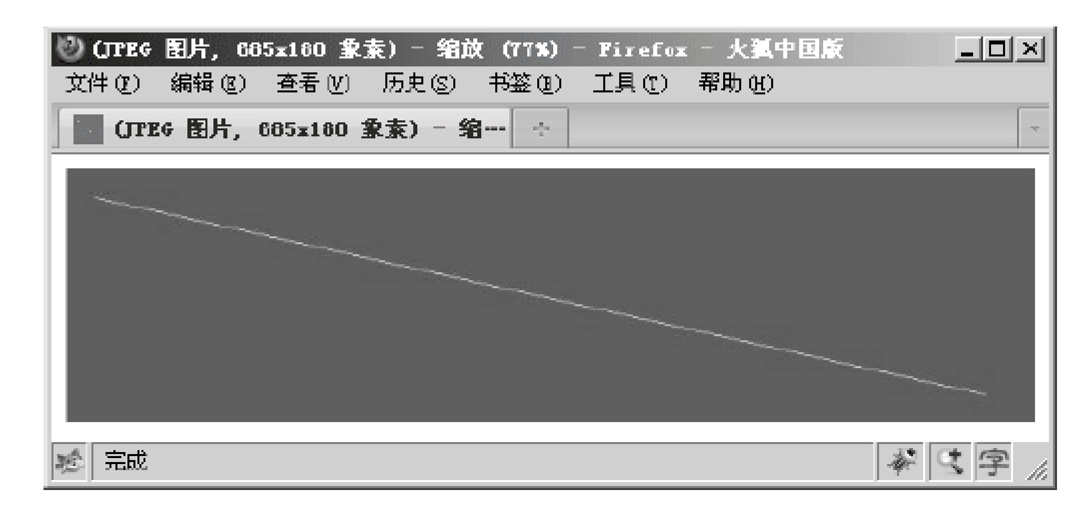


图 12.4 绘制一条白色直线

12.3.3 绘制文字

应用 PHP 中的 GD 库既可以绘制英文字符串，也可以绘制中文汉字。绘制英文字符串应用 imagestring()函数，语法如下：

```
bool imagestring ( resource image, int font, int x, int y, string s, int col )
```

imagestring()函数用 col 颜色将字符串 s 绘制到 image 所代表的图像的 x,y 坐标处（这是字符串左上角坐标，整幅图像的左上角为 0, 0）。如果 font 是 1、2、3、4 或 5，则使用内置字体。

绘制中文汉字应用 imagettftext()函数，语法如下：

```
array imagettftext ( resource image, float size, float angle, int x, int y, int color, string fontfile, string text )
```

imagettftext()函数的参数说明如表 12.1 所示。

表 12.1 imagettftext()函数的参数说明

参数	说明
image	图像资源
size	字体大小。根据 GD 版本不同，应该以像素大小指定（GD1）或点大小（GD2）
angle	字体的角度，顺时针计算，0°为水平，也就是 3 点钟的方向（由左到右），90°则为由下到上的文字

续表

参　数	说　明
x	文字的 x 坐标值。它设定了第一个字符的基本点
y	文字的 y 坐标值。它设定了字体基线的位置，不是字符的最底端
color	文字的颜色
fontfile	字体的文件名称，也可以是远端的文件
text	字符串内容

技巧： 在 GD2 函数库中支持的是 UTF-8 编码格式的中文，所以在通过 imagettftext()函数输出中文字符串时，必须保证中文字符串的编码格式是 UTF-8，否则中文将不能正确地输出。如果定义的中文字符串是 GB2312 简体中文编码，那么要通过 iconv()函数对中文字符串的编码格式进行转换。

例 12.03　通过 imagestring()函数水平地绘制一行字符串“I like PHP”。（**实例位置：光盘\mr\12\sl\12.03**）

首先，创建一个画布。然后，定义画布的背景颜色和输出字符串的颜色。接着，通过 imagestring()函数水平地绘制一行英文字符串。最后，输出图像并且销毁图像资源。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");          //设置页面的编码风格
    header("Content-Type:image/jpeg");                        //告知浏览器输出的是一张图片
    $image=imagecreate(300,80);                               //创建画布的大小
    $bgcolor=imagecolorallocate($image,200,60,90);            //设置背景颜色
    $write=imagecolorallocate($image,0,0,0);                  //设置文字颜色
    imagestring($image,5,80,30,"I Like PHP","$write");       //书写英文字符
    imagejpeg($image);                                        //输出图像
    imagedestroy($image);                                     //销毁图像
?>
```

运行结果如图 12.5 所示。



图 12.5　绘制英文字符串

例 12.04　通过 imagettftext ()函数水平地绘制一行中文字符串。（**实例位置：光盘\mr\12\sl\12.04**）

首先，创建一个画布，定义画布背景颜色和输出字符串的颜色。然后，定义中文字符串使用的字体，以及要输出的中文字符串的内容。接着，通过 imagettftext()函数水平地绘制一行中文字符串。最后，输出图像并且销毁图像资源。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");               //设置文件编码格式
    header("Content-Type:image/jpeg");                             //告知浏览器所要输出图像的类型
    $image=imagecreate(800,150);                                   //创建画布
    $bgcolor=imagecolorallocate($image,0,200,200);                 //设置图像背景色
    $fontcolor=imagecolorallocate($image,200,80,80);               //设置字体颜色为黑色
    $font="FZSHHJW.TTF";                                           //定义字体
```

```
    $string="明日科技";                                    //定义输出中文
    imagettftext($image,80,5,100,130,$fontcolor,$font,$string);  //写 TTF 文字到图中
    imagejpeg($image);                                     //建立 JPEG 图形
    imagedestroy($image);                                  //释放内存空间
?>
```

运行结果如图 12.6 所示。

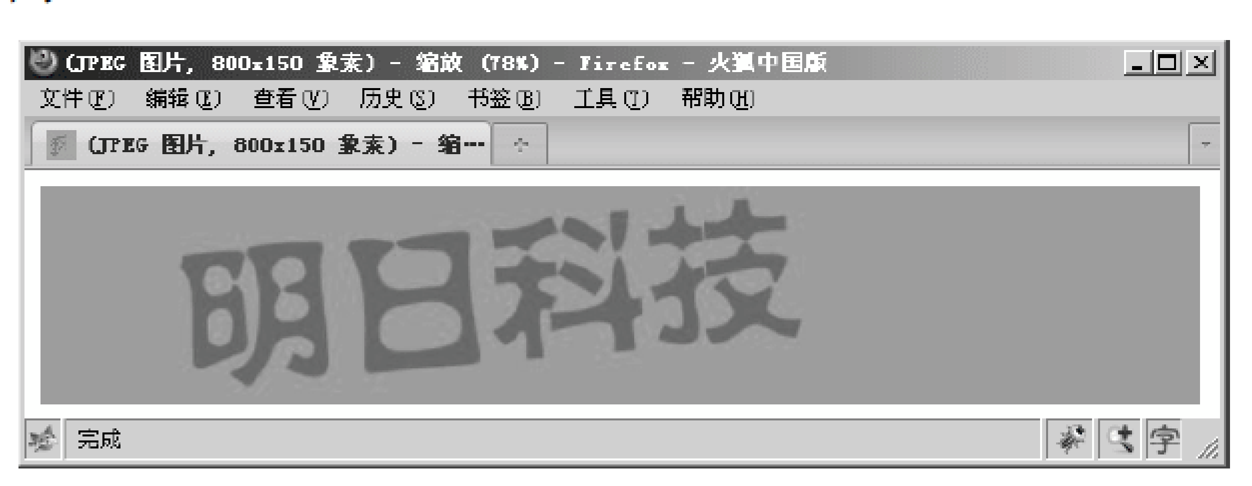


图 12.6　绘制中文字符串

提示：由于 imagettftext()函数只支持 UTF-8 编码，如果创建的网页的编码格式使用 GB2312，那么在应用 imagettftext()函数输出中文字符串时，必须应用 iconv()函数将字符串的编码格式由 GB2312 转换为 UTF-8，否则在输出时将会显示乱码。在本实例中之所以没有进行编码格式转换，是因为创建的文件默认使用的是 UTF-8 编码。

12.3.4　输出图像

PHP 作为一种 Web 语言，无论是解析出的 HTML 代码还是二进制的图片最终都要通过浏览器显示。应用 GD2 函数绘制的图像首先需要用 header()函数发送 HTTP 头信息给浏览器，告知所要输出图像的类型，然后应用 GD2 函数库中的函数完成图像输出。

header()函数用于向浏览器发送 HTTP 头信息，语法如下：

```
void header ( string string [, bool replace [, int http_response_code]] )
```

参数说明：

- ☑ string：发送的标头。
- ☑ replace：如果一次发送多个标头，对于相似的标头是替换还是添加。如果是 FALSE，则强制发送多个同类型的标头。默认为 TRUE，即替换。
- ☑ http_response_code：强制 HTTP 响应为指定值。

使用 header()函数可以实现如下 4 种功能。

（1）重定向，这是最常用的功能。

```
header("Location: http://www.mrbccd.com");
```

（2）强制客户端每次访问页面时获取最新资料，而不是使用存在于客户端的缓存。

```
//设置页面的过期时间(用格林威治时间表示)。
header("Expires:    Mon,    08    Jul    2008    08:08:08    GMT");
//设置页面的最后更新日期(用格林威治时间表示)，使浏览器获取最新资料
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "GMT");
header("Cache-Control: no-cache,    must-revalidate");  //控制页面不使用缓存
header("Pragma: no-cache");                             //参数（与以前的服务器兼容），即兼容 HTTP 1.0 协议
header("Content-type:    application/file");            //输出 MIME 类型
header("Content-Length:    227685");                    //文件长度
header("Accept-Ranges:    bytes");                      //接受的范围单位
```

```
//默认时文件保存对话框中的文件名称
header("Content－Disposition: attachment; filename=$filename");    //实现下载
```

（3）输出状态值到浏览器，控制访问权限。

```
header('HTTP/1.1 401 Unauthorized');
header('status: 401 Unauthorized');
```

（4）完成文件的下载。

```
header("Content-type: application/x-gzip");
header("Content-Disposition: attachment; filename=文件名");
header("Content-Description: PHP3 Generated Data"); >
```

在应用的过程中，唯一需要改动的就是 filename。即将 filename 替换为要下载的文件。

imagegif()函数以 GIF 格式将图像输出到浏览器或文件，语法如下：

```
bool imagegif ( resource image [, string filename] )
```

参数说明：

☑ image：是 imagecreate()或 imagecreatefromgif()等创建图像函数的返回值，图像格式为 GIF。如果应用 imagecolortransparent()函数，则使图像设置为透明，图像格式为 GIF。

☑ filename：可选参数，如果省略，则原始图像流将被直接输出。

imagejpeg()和 imagepng()函数的使用方法与 imagegif()函数类似，这里不再赘述。至于图像输出函数的应用在前面的小节中都已经使用过，这里不再重新举例。

12.3.5　销毁图像

在 GD2 函数库中通过 imagedestroy()函数来销毁图像，释放内存。其语法如下：

```
bool imagedestroy ( resource image )
```

通过 imagedestroy()函数释放与 image 关联的内存。image 是由图像创建函数返回的图像标识符，如 imagecreatetruecolor()。

有关销毁图像函数的应用在前面的小节中都已经使用过，这里不再重新举例。

12.4　运用 Jpgraph 类库绘制图像

Jpgraph 是基于 GD2 函数库编写的主要用于创建统计图的类库。在绘制统计图方面不仅功能非常强大，而且代码编写方便，只需简单的几行代码就可以绘制出非常复杂的统计图效果，从而在很大程度上提高了编程人员的开发效率。

12.4.1　Jpgraph 类库简介

Jpgraph 类库是一个可以应用在 PHP 4.3.1 以上版本的用于图形图像绘制的类库，该类库完全基于 GD2 函数库编写，Jpgraph 类库提供了多种方法创建各类统计图，包括坐标图、柱状图、饼形图等。使用 Jpgraph 类库使复杂的统计图编写工作变得简单，大大提高了开发者的开发效率，在现今的 PHP 项目中被广泛应用。

12.4.2　Jpgraph 的安装

安装 Jpgraph 前，首先需要下载该类库的压缩包，Jpgraph 类库的压缩包主要有两种形式，即 zip 格式和

tar 格式。如果是 Linux/UNIX 平台，可以选择 tar 格式的压缩包；如果是微软的 win32 平台，选择上述两种格式的压缩包任一种即可。Jpgraph 类库可以从其官方网站“http://www.aditus.nu/jpgraph/”下载，目前最新的版本是 2.3.4。

如果已经下载 Jpgraph 的安装包，解压后将会呈现如图 12.7 所示的目录结构。

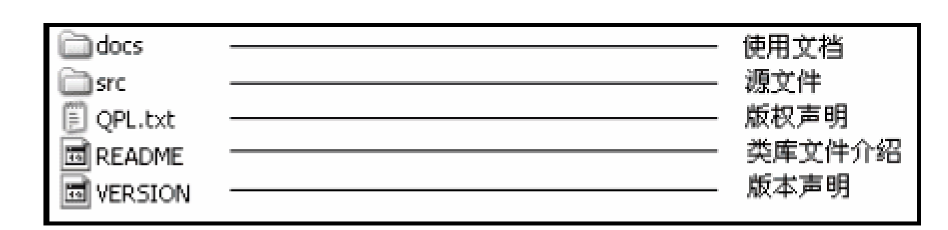


图 12.7　Jpgraph 压缩包解压后的目录

（1）如果希望服务器中所有站点均有效，可以按如下步骤进行配置。

首先，解压下载的压缩包，复制图 12.7 中的 src 文件夹，并将该文件夹保存到服务器磁盘中，如 c:\jpgraph。

然后，编辑 php.ini 文件，修改 include_path 配置项，在该项后增加 Jpgraph 类库的保存目录，如 include_path = “.;c:\jpgraph”。

最后，重新启动 Apache 服务，则配置生效。

（2）如果只希望在本站点使用 Jpgrahp，则直接将 src 文件夹复制到工程目录下即可。

通过上述两种方式都可以完成 Jpgraph 的安装，此时在程序中通过 require_once 语句即可完成 Jpgraph 类库的载入操作。代码如下：

```
require_once 'src/jpgraph.php';
```

说明： 因为 Jpgraph 类库属于第三方的内容，所以在本书光盘中没有提供。运行本堂课中所有涉及 Jpgraph 类库的程序时，都需要读者自己下载 Jpgraph 类库，然后复制 src 文件夹将其放置于实例根目录的上级文件夹下。

12.4.3　Jpgraph 创建 3D 饼形图展示部门业绩比较

例 12.05　使用 Jpgraph 类库制作统计图功能极其强大，不仅可以绘制平面图形，而且可以绘制具有 3D 效果的图形。直接使用 GD2 函数库可以绘制出各种图形，当然也包括 3D 饼形图，但使用 GD2 函数绘制 3D 图形需要花费大量的时间，而且相对复杂，而采用 Jpgraph 类库绘制 3D 饼形图却十分方便、快捷。**（实例位置：光盘\mr\12\sl\12.05）**

具体步骤如下：

（1）在程序中导入 Jpgraph 类库及饼图绘制功能。代码如下：

```
include ("../src/jpgraph.php");              //导入 Jpgraph 类库
include ("../src/jpgraph_pie.php");          //导入 Jpgraph 类库的饼图功能
include ("../src/jpgraph_pie3d.php");        //导入 Jpgraph 类库的 3D 饼形图功能
```

（2）创建数值型数组作为统计数据。代码如下：

```
$data = array(
    array(80,18,15,17),
    array(35,28,6,34),
    array(10,28,10,5),
    array(22,22,10,17));                     //设置统计数据
```

（3）创建统计图对象，并对统计图的标题内容、字体进行设置。代码如下：

```
$graph->title->Set(iconv('utf-8','gbk',"部门业绩比较"));          //设置标题
$graph->title->SetFont(FF_SIMSUN,FS_BOLD,20);                    //设置字体
$graph->title->SetColor('white');                                //设置颜色
```

（4）创建 3D 饼图对象并输入统计图。代码如下：

```
$titles = array('C#','.net','JAVA','PHP');
$n = count($piepos)/2;
$graph = new PieGraph(550,400,'auto');
// Specify margins since we put the image in the plot area
$graph->SetMargin(1,1,40,1);
$graph->SetMarginColor('navy');
$graph->SetShadow(false);
// Setup background
$graph->SetBackgroundImage('worldmap1.jpg',BGIMG_FILLPLOT);
```

运行结果如图 12.8 所示。

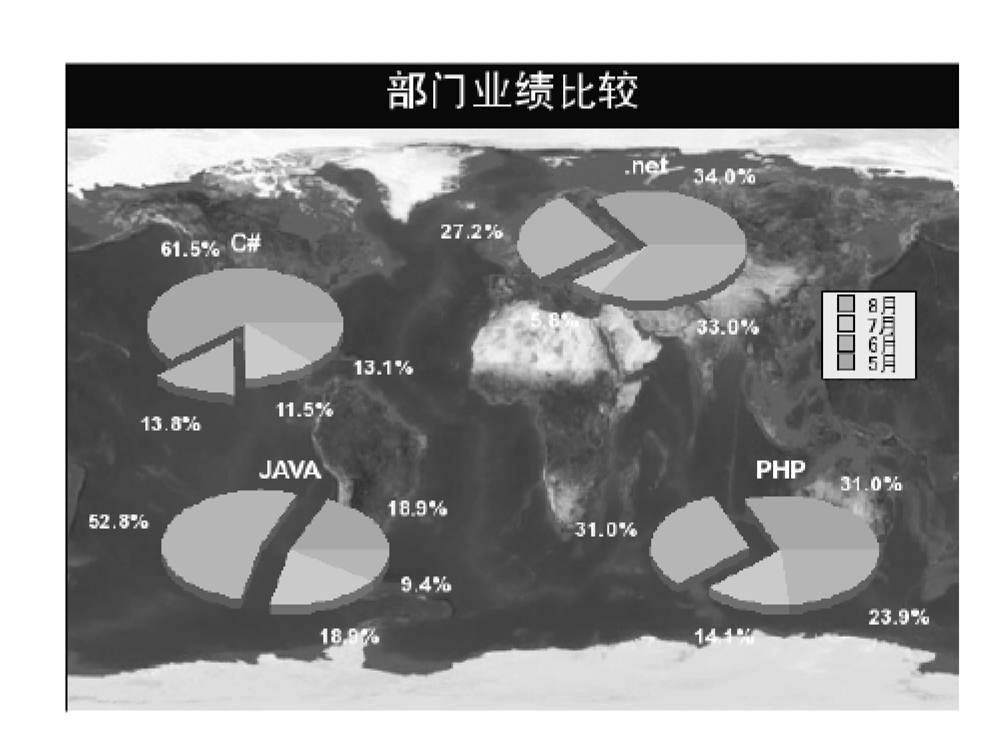


图 12.8　部门业绩比较

12.5　照猫画虎——基本功训练

12.5.1　基本功训练 1——绘制一个正方形

视频讲解：光盘\mr\12\lx\绘制一个正方形.exe

实例位置：光盘\mr\12\zmhh\01

利用 GD2 函数绘制一个正方形图像其实是非常简单的。只需要设置图像的 X 轴和 Y 轴长度相等即可。本实例运行结果如图 12.9 所示。

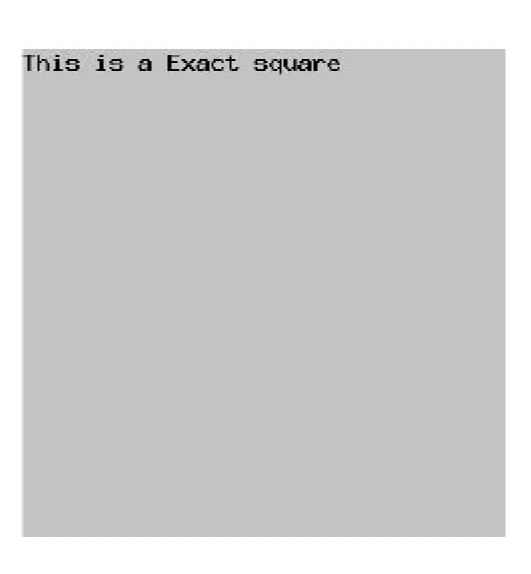


图 12.9　绘制一个正方形

操作步骤如下：

创建脚本文件 index.php。首先，设置页面的编码风格。然后，告知浏览器输出的图像格式为 image/jpeg。

最后，利用 GD2 函数绘制图像，其中使用的函数包括 imagecreate()函数、imagecolorallocate()函数、imagestring()函数、imagejpeg()函数和 imagedestroy()函数等。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                    //设置编码格式
    header("Content-Type:image/jpeg");                                  //告知浏览器输出的是一个图像
    $image=imagecreate(300,300);                                        //创建图像
    $bgcolor=imagecolorallocate($image,200,80,100);                     //设置背景颜色
    $fontcolor=imagecolorallocate($image,0,0,0);                        //设置字体颜色
    imagestring($image,5,0,0,"This is a Exact square",$fontcolor);      //向图像中写入字母
    imagejpeg($image);                                                  //输出图像
    imagedestroy($image);                                               //销毁图像
?>
```

照猫画虎：绘制一个长方形图像。（20 分）（实例位置：光盘\mr\12\zmhh\01_zmhh）

12.5.2 基本功训练 2——在画布中输出一行英文字符串

视频讲解：光盘\mr\12\lx\在画布中输出一行英文字符串.exe

实例位置：光盘\mr\12\zmhh\02

强大的 GD2 函数库几乎可以满足用户的所有图像需求，比较常用的验证码操作就是一个非常典型的例子。本实例是一个非中文验证码的雏形。希望用户究其根本，使原理熟烂于心中。本实例的运行结果如图 12.10 所示。

A B C D E F

图 12.10 输出一行英文字符串

实现本实例首先是通过 header()函数告知浏览器本操作输出的是一个图像。然后利用 imagecreate()函数创建图像，利用 imagecolorallocate()函数设置背景色和字体颜色，利用 imagestring()函数输出英文字符串。最后利用 imagejpeg()函数输出图像。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                    //设置字符集
    header("Content-Type:image/jpeg");                                  //告知浏览器输出的是一个图像
    $image=imagecreate(130,30);                                         //设置画布大小
    $bgcolor=imagecolorallocate($image,230,230,230);                    //设置背景颜色
    $fontcolor=imagecolorallocate($image,70,70,70);                     //设置字体颜色
    imagestring($image,5,16,6,"A B C D E F",$fontcolor);                //输出英文字符串
    imagejpeg($image);                                                  //输出图像
    imagedestroy($image);                                               //销毁图像
?>
```

照猫画虎：改变输出本实例的中英文字母的大小。（20 分）（实例位置：光盘\mr\12\zmhh\02_zmhh）

12.5.3 基本功训练 3——在背景图像上输出北京欢迎您

视频讲解：光盘\mr\12\lx\在背景图像上输出北京欢迎您.exe

实例位置：光盘\mr\12\zmhh\03

PHP 中的 GD2 库支持中文，但是必须要以 UTF-8 格式进行传递，如果直接绘制中文字符串就会显示乱码，这是因为 GD2 对中文只能接收 UTF-8 编码格式，并且默认使用了英文的字体，所以只需将显示的中文字符串进行转码，并设置绘制中文字符使用的字体，即可绘制中文字符。在 GD2 函数库中应用 imagestring()

函数在画布上输出字符串。本实例运行结果如图 12.11 所示。

图 12.11　背景图像上输出北京欢迎您

操作步骤如下：

首先，通过 header()函数告知浏览器本操作输出的是一个图像。然后，利用 imagecreate()函数创建图像，利用 imagecolorallocate()函数设置背景色和字体颜色，利用 imagefttext()函数输出中文字符串。最后利用 imagejpeg()函数输出图像。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");          //设置字符集
    header("Content-Type:image/jpeg");                       //告知浏览器输出的是一个图像
    $image=imagecreate(300,100);                             //创建画布
    $bgcolor=imagecolorallocate($image,250,180,180);         //设置背景颜色
    $fontcolor=imagecolorallocate($image,30,30,30);          //设置字体颜色
    $font="STCAIYUN.TTF";                                    //设置字体
    $string="北京欢迎您";                                     //设置文本
    imagefttext($image,30,5,40,60,$fontcolor,$font,$string);
    imagejpeg($image);
    imagedestroy($image);
?>
```

照猫画虎：更改本实例中文字的倾斜角度并且输出同样的文字内容。（20 分）（实例位置：光盘\mr\12\zmhh\03_zmhh）

12.5.4　基本功训练 4——在照片上书写文字

视频讲解：光盘\mr\12\lx\在照片上书写文字.exe

实例位置：光盘\mr\12\zmhh\04

为保证网站中所上传的图片不被他人盗用，经常需要在所上传的图片中添加水印。在 PHP 中使用 imageTTFText()函数即可将文字动态地添加到图片中。本实例的运行结果如图 12.12 所示。



图 12.12　在照片上书写文字

程序开发步骤如下：

（1）使用 header()函数发送 HTTP 头指定发送媒体类型。

（2）使用 iconv()函数将“编程词典是您最好的选择”由 GB2312 格式转换为 UTF-8 格式。

（3）载入一张 JPG 格式的背景图片，并用 imageTTFText()函数在图像中以 STCAIYUN.TTF 字体输出

一串黑色的文字“编程词典是您最好的选择”。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                 //设置字符集
    header("Content-Type:image/jpeg");                              //告知浏览器输出的是一个图像
    $image=imagecreatefromjpeg("gg.jpg");                           //载入图片
    $fontcolor=imagecolorallocate($image,0,0,0);                    //设置字体颜色
    $font="STCAIYUN.TTF";                                           //设置字体
    $string="编程词典是您最好的选择";                                  //设置字符串内容
    imagefttext($image,30,5,30,260,$fontcolor,$font,$string);
    imagejpeg($image);
    imagedestroy($image);
?>
```

照猫画虎：更改图片上的文字内容为明日科技。（20 分）（实例位置：光盘\mr\12\zmhh\04_zmhh）

12.5.5 基本功训练 5——Jpgraph 创建折线图分析图书销售走势

视频讲解：光盘\mr\12\lx\Jpgraph 创建折线图分析图书销售走势.exe

实例位置：光盘\mr\12\zmhh\05

在企业运营中，经常需要对各种数据进行统计，利用图表动态分析不同的数据表中的数据，可以帮助企业领导者更好地制定决策，而且可以使数据显示更加直观。本实例的运行结果如图 12.13 所示。

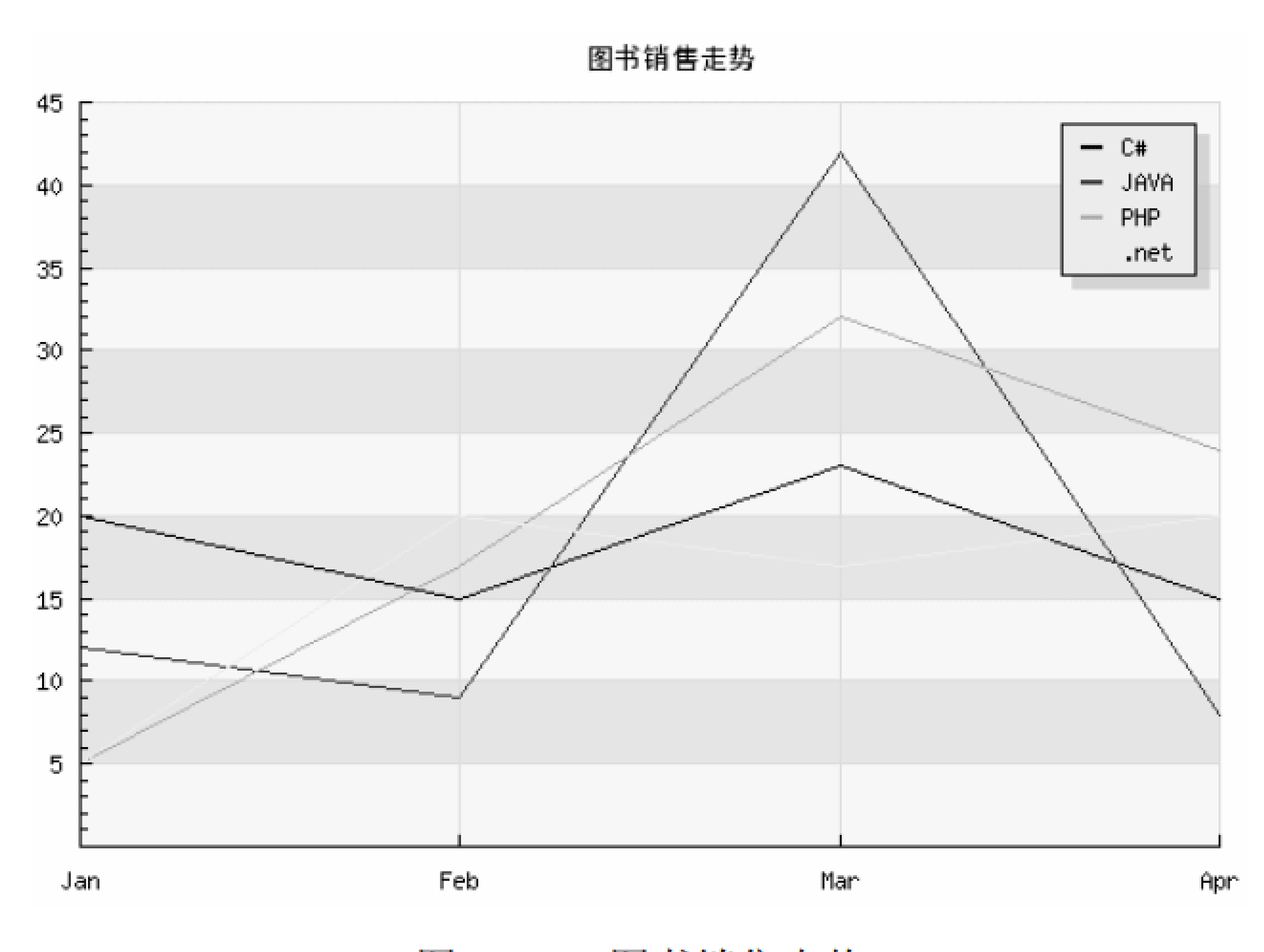


图 12.13 图书销售走势

具体实现过程如下：

（1）在程序中导入 Jpgraph 类库及直线绘制功能。代码如下：

```
include ("../src/jpgraph.php");                    //导入 Jpgraph 类库
include ("../src/jpgraph_line.php");               //导入 Jpgraph 类库的柱状图功能
```

（2）创建统计数据，并将数据保存到一维数组中。代码如下：

```
$datay1 = array(20,15,23,15);                      //设置统计数据
$datay2 = array(12,9,42,8);
$datay3 = array(5,17,32,24);
$datay4 = array(5,20,17,20);
```

（3）创建统计图对象，并设置坐标轴的刻度样式。代码如下：

```
$graph = new Graph(600,400);                    //创建统计图对象
$graph->SetMarginColor('white');
$graph->SetScale("textlin");                    //设置刻度样式
$graph->SetFrame(false);
$graph->SetMargin(30,50,30,30);
```

照猫画虎：将图 12.13 中的红色线条的标题更改为 VC。(20 分)(实例位置：光盘\mr\12\zmhh\05_zmhh)

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

12.6　情景应用——拓展与实践

12.6.1　情景应用 1——GD2 函数生成图像验证码

视频讲解：光盘\mr\12\lx\GD2 函数生成图像验证码.exe

实例位置：光盘\mr\12\qjyy\01

用户注册主要就是为了保存用户的一些基本信息，包括用户名、真实姓名、密码等，以及跟踪和定位不同用户，从而更好地服务于用户。同时为了提高站点的安全性，避免由于网速慢造成用户注册信息的重复提交，往往会在用户注册表单中增加验证码功能。本实例的运行结果如图 12.14 所示。



图 12.14　生成图像验证码

操作步骤如下：

（1）设置页面的编码风格为 utf-8。

（2）告知浏览器输出的是一个 jpeg 格式的图像。

（3）利用相关函数编写验证码。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                 //设置编码风格
    header("Content-Type:image/jpeg");                              //设置图片格式
    $image=imagecreate(250,100);                                    //创建画布
    $bgcolor=imagecolorallocate($image,250,180,180);                //设置背景颜色
    $fontcolor=imagecolorallocate($image,30,30,30);                 //设置字体颜色
    $font="STCAIYUN.TTF";                                           //设置字体
    for($a=0;$a<4;$a++){                                            //循环语句
        $rand.=dechex(rand(0,15));
    }
    $string=$rand;
    imagefttext($image,50,7,40,80,$fontcolor,$font,$string);        //输出验证码
    imagejpeg($image);
    imagedestroy($image);
?>
```

DIY：在图 12.14 中的文本中添加一个中文文字。(10 分)(实例位置：光盘\mr\12\qjyy\01_diy)

12.6.2 情景应用 2——通过图像显示密码安全强度

视频讲解：光盘\mr\12\lx\通过图像显示密码安全强度.exe

实例位置：光盘\mr\12\qjyy\02

密码安全强度是表单注册页面常有的模块，它的主要作用是提醒用户密码是否足够复杂和安全，其优点体现在两方面：一方面高强度的密码是用户注册信息的保证，另一方面也是表单注册页面的一个亮点。本实例是密码安全强度的一个简单的结构，用户可以根据情况扩展相关知识。本实例的运行结果如图 12.15 所示。



图 12.15 密码安全强度

操作步骤如下：

（1）设置页面的编码风格为 utf-8。

（2）编写表单，设置“安全测试”按钮，利用 JavaScript 代码实现当“安全测试”按钮被单击时，根据文本框提供的信息显示相应的图像信息。当文本框信息小于 4 个字节时，显示密码安全强度低；当文本框信息大于 4 个字节时，显示密码安全强度高。代码如下：

```
<div id="body">                          //设置表单
        <input class="e" type="pwd"><span></span> <input class="f"type="submit" value="安全测试">
</div>
<table class=a>
    <tr>
        <td>密码强度为低</td>
    </tr>
</table>
<table class=c>
    <tr>
        <td>密码强度为高</td>
    </tr>
</table>
<script>
    $(function(){                                        //JavaScript 代码
        $("input:eq(1)").click(function(){               //当“安全测试”按钮被单击时
            var aq=$("input:first").val();               //获取文本框信息
            if(aq==""){                                  //如果文本框信息为空
                alert("密码框不能为空");                  //输出提示
            }else{
                if(aq.length<4){                         //当文本框信息小于 4 个字节
                    $("span").text("");
                    $("span").text($(".a").css("display","block"));        //更改表格 1 的 css 属性
                }else{
                    $("span").text("");
                    $("span").text($(".c").css("display","block"));        //否则更改表格 2 的 css 属性
                }
            }
        });
    });
</script>
```

DIY：在图 12.15 的安全强度中当文本框内容为空时输出强度提示。（10 分）（实例位置：光盘\mr\12\qjyy\02_diy）

12.6.3　情景应用 3——通过柱状图展示 2010 年第一季度编程词典销量

视频讲解：光盘\mr\12\lx\通过柱状图展示 2010 年第一季度编程词典销量.exe

实例位置：光盘\mr\12\qjyy\03

在本实例中通过多柱形图展示 2010 年第一季度编程词典销量，其中展示 C#、VB、VC、Java 和 ASP.NET 5 个语言 2010 年第一季度的销量。本实例的运行结果如图 12.16 所示。

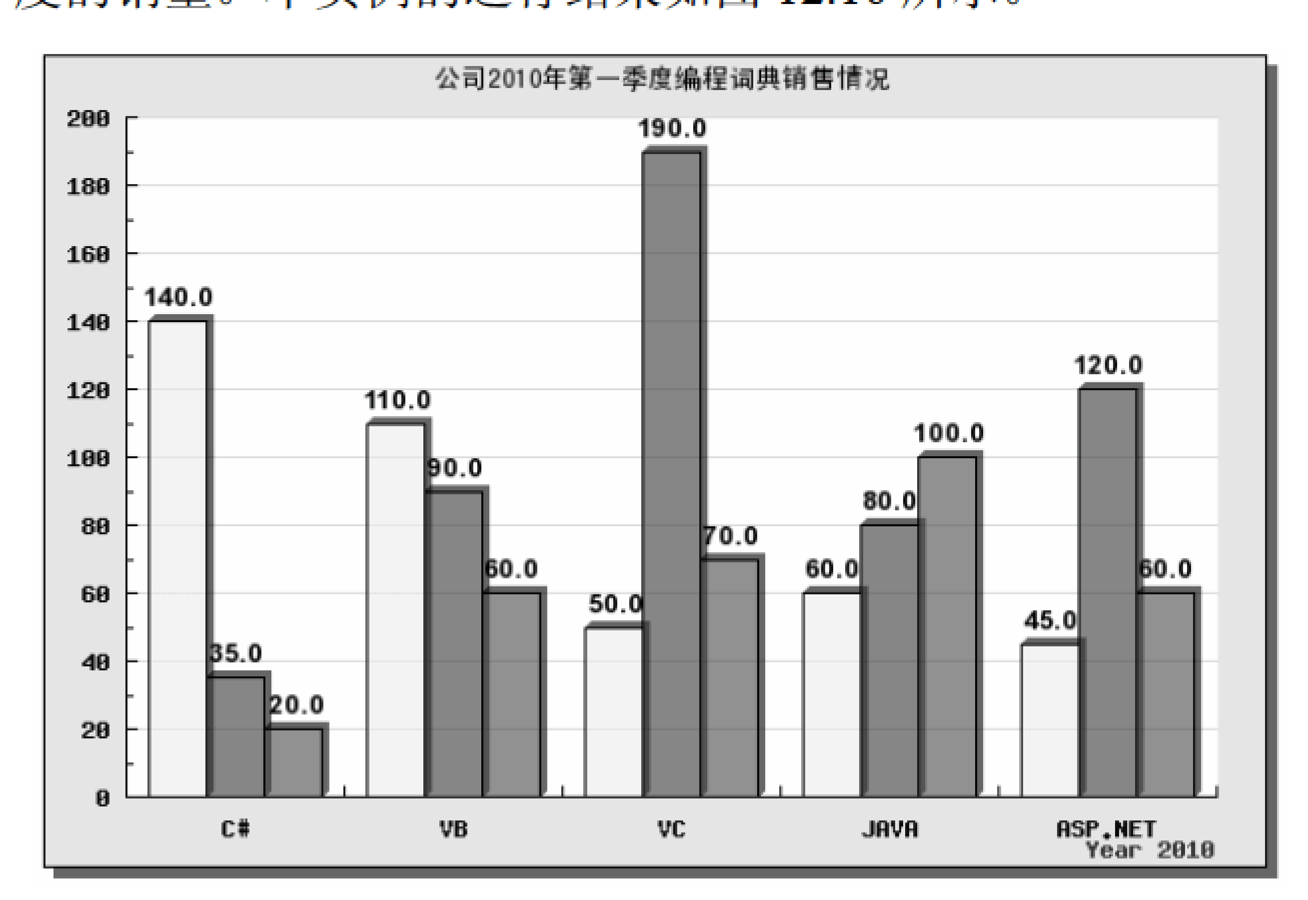


图 12.16　通过柱状图展示 2010 年第一季度编程词典销量

具体步骤如下：

（1）创建 index.php 文件，设置网页的编码格式，并通过 require_once()语句导入 Jpgraph 类库。

（2）应用 Jpgraph 类库中的方法创建一个柱状图数组，对 2010 年第一季度编程词典销量进行分析。其关键是将 BarPlot 类创建的单柱状图通过 GroupBarPlot 类整合到同一个图像中进行输出。

☑　使用 BarPlot 类创建统计图的柱状效果。

通过 Jpgraph 创建柱状图需要使用 BarPlot 类，该类的构造方法中包含一个数值数组参数，该数组的内容即为需要统计的数据。BarPlot 类的使用格式如下：

```
$barPlot = new BarPlot($data)
```

参数$data 是由统计数据所组成的数组。

☑　使用 GroupBarPlot 类创建多个柱状效果。

通过 Jpgraph 创建多个柱状图需要使用 GroupBarPlot 类，该类的构造方法中包含一个数组参数，该数组的内容即为需要整合的单个柱状图对象。GroupBarPlot 类的语法如下：

```
$gbarplot= new GroupBarPlot ($data)
```

参数$data 是由单个柱状图对象所组成的数组。

完整代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
require_once ("../src/jpgraph.php");                          //载入类库文件
require_once ("../src/jpgraph_bar.php");                      //载入类库文件
```

```
require_once ("../src/jpgraph_flags.php");              //载入类库文件
$datay1=array(140,110,50,60,45);                        //定义一月份数据
$datay2=array(35,90,190,80,120);                        //定义二月份数据
$datay3=array(20,60,70,100,60);                         //定义三月份数据
$graph = new Graph(600,400);                            //创建图像
$graph->SetScale("textlin");                            //设置刻度值类型
$graph->SetMargin(40,30,30,40);                         //定义边框距离
$graph->SetMarginColor('white:0.9');                    //设置图像边框颜色
$graph->SetColor('white');                              //设置图像的填充颜色
$graph->SetShadow();                                    //设置阴影
$datas = array("C#", "VB", "VC", "JAVA", "ASP.NET");    //设置统计数据
$graph->xaxis->SetTickLabels($datas);
$graph->xaxis->title->Set('Year 2010');                 //设置 X 轴标题
$graph->xaxis->title->SetFont(FF_FONT1,FS_BOLD);        //设置标题字体
$graph->xaxis->title->SetColor('red');                  //设置字体颜色
$graph->xaxis->SetFont(FF_FONT1,FS_BOLD);               //设置 X 轴字体
$graph->xaxis->SetColor('navy');                        //设置 X 轴刻度线颜色
$graph->yaxis->SetFont(FF_FONT1,FS_BOLD);               //设置 Y 轴字体
$graph->yaxis->SetColor('navy');                        //设置 Y 轴刻度线颜色
$graph->title->Set(iconv("utf-8","gb2312",'公司 2010 年第一季度编程词典销售情况'));        //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);             //设置标题字
$graph->title->SetMargin(3);                            //设置图像距离顶部距离
//创建 3 个方柱统计图
$bplot1 = new BarPlot($datay1);
$bplot2 = new BarPlot($datay2);
$bplot3 = new BarPlot($datay3);
//填充图像颜色
$bplot1->SetFillColor('yellow@0.4');
$bplot2->SetFillColor('red@0.4');
$bplot3->SetFillColor('darkgreen@0.4');
//设置图像的阴影效果
$bplot1->SetShadow('black@0.4');
$bplot2->SetShadow('black@0.4');
$bplot3->SetShadow('black@0.4');
$bplot1->value->Show();                                 //输出图像对应的数据值
$bplot1->value->SetFont(FF_ARIAL,FS_BOLD);              //定义图像值的字体
$bplot1->value->SetColor("black","darkred");            //定义值的颜色
$bplot2->value->Show();                                 //输出图像对应的数据值
$bplot2->value->SetFont(FF_ARIAL,FS_BOLD);              //定义图像值的字体
$bplot2->value->SetColor("black","darkred");            //定义值的颜色
$bplot3->value->Show();                                 //输出图像对应的数据值
$bplot3->value->SetFont(FF_ARIAL,FS_BOLD);              //定义图像值的字体
$bplot3->value->SetColor("black","darkred");            //定义值的颜色
//实例化图像操作组类
$gbarplot = new GroupBarPlot(array($bplot1,$bplot2,$bplot3));
$gbarplot->SetWidth(0.8);                               //设置图像大小
$graph->Add($gbarplot);                                 //添加数据
$graph->Stroke();                                       //生成图像
?>
```

DIY：改变柱状图，仍然展示编程词典的销量。（20 分）（实例位置：光盘\mr\12\qjyy\03_diy）

12.6.4　情景应用 4——通过折线图分析 2010 年公司销售额

视频讲解：光盘\mr\12\lx\通过折线图分析 2010 年公司销售额.exe

实例位置：光盘\mr\12\qjyy\04

在企业运营中，经常需要对各种数据进行统计，利用图表动态分析不同的数据表中的数据。这可以帮助企业领导者更好地制定决策，还可以使数据显示更加直观。本实例利用 Jpgraph 类库实现 2010 年公司销售额的折线图，运行结果如图 12.17 所示。

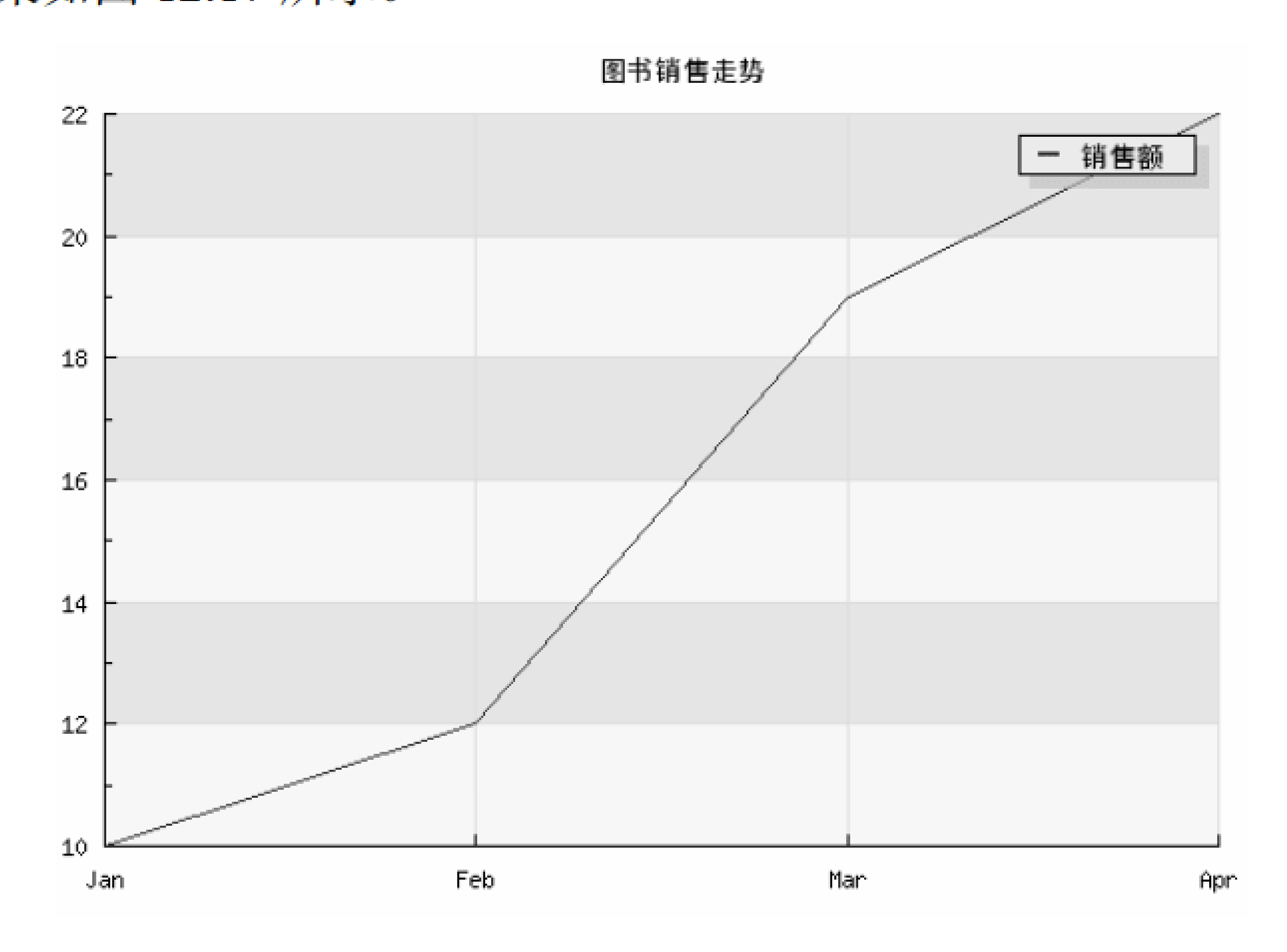


图 12.17　2010 年公司销售额

具体过程如下：

（1）在程序中导入 Jpgraph 类库及直线绘制功能，代码如下：

```
include ("../src/jpgraph.php");                    //导入 Jpgraph 类库
include ("../src/jpgraph_line.php");               //导入 Jpgraph 类库的柱状图功能
```

（2）创建统计数据，并将数据保存到一维数组中，代码如下：

```
$datay1 = array(10,12,19,22);                      //设置统计数据
```

（3）创建统计图对象，并设置坐标轴的刻度样式。代码如下：

```
$graph = new Graph(600, 300);                      //创建统计图对象
$graph->SetScale("textlin");                       //设置刻度样式
$graph->setY2Scale("lin");
$graph->SetShadow();                               //设置背景带阴影
$graph->img->SetMargin(40, 50, 20, 70);            //设置统计图边距范围
$graph->title->Set("图书销售走势");                //设置统计图标题
```

（4）创建 LinePlot 对象，并将创建后的对象添加到统计图对象中。代码如下：

```
$lineplot1 = new LinePlot($data1);                 //建立 LinePlot 对象
$graph->Add($lineplot1);
$graph->AddY2($lineplot2);
```

（5）设置统计图和坐标轴的标题内容和文字样式等，并输出统计图。代码如下：

```
$graph->xaxis->title->Set("月份");                 //设置横坐标名称
$graph->yaxis->title->Set("book A 销售金额（万元）");   //设置左侧纵坐标名称
$graph->y2axis->title->Set("book B 销售金额（万元）");  //设置右侧纵坐标名称
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);
```

```
$graph->yaxis->title->SetFont(FF_SIMSUN, FS_BOLD);
$graph->y2axis->title->SetFont(FF_SIMSUN, FS_BOLD);
$graph->xaxis->title->SetFont(FF_SIMSUN, FS_BOLD);              //设置字体
```

DIY：在图 12.17 的图书销售走势中将红色的线条更改为黄色。（20 分）（实例位置：光盘\mr\12\qjyy\04_diy）

12.6.5　情景应用 5——通过饼形图区块分析 2010 年图书销量

视频讲解：光盘\mr\12\lx\通过饼形图区块分析 2010 年图书销量.exe

实例位置：光盘\mr\12\qjyy\05

用 Jpgraph 类库制作统计图功能极其强大，不仅可以绘制平面图形，而且可以绘制具有 3D 效果的图形。12.4.3 小节中已经对 3D 的效果图形进行了介绍，这里不再赘述。本实例的运行结果如图 12.18 所示。

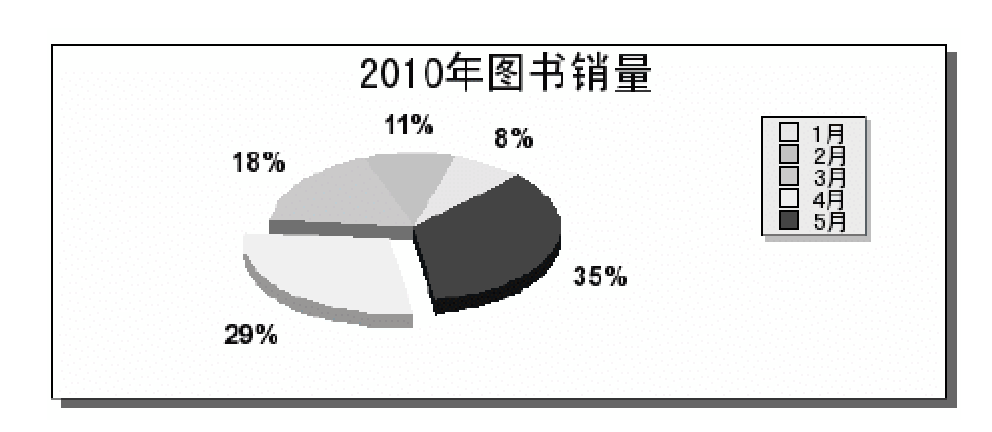


图 12.18　2010 年图书销量

具体步骤如下：

（1）在程序中导入 Jpgraph 类库及饼图绘制功能，代码如下：

```
include ("../src/jpgraph.php");                    //导入 Jpgraph 类库
include ("../src/jpgraph_pie.php");                //导入 Jpgraph 类库的饼图功能
include ("../src/jpgraph_pie3d.php");              //导入 Jpgraph 类库的 3D 饼形图功能
```

（2）创建数值型数组作为统计数据，代码如下：

```
$data = array(20,27,45,75,90);                     //设置统计数据
```

（3）创建统计图对象，并对统计图的标题内容、字体进行设置。代码如下：

```
$graph->title->Set(iconv('utf-8','gbk',"2010 年图书销量"));  //设置标题的字体和编码风格
$graph->title->SetFont(FF_SIMSUN,FS_BOLD,18);
$graph->title->SetColor("darkblue");
$graph->legend->Pos(0.1,0.2);
```

DIY：在图 12.18 的图书销量中将 5 月的销量更改为 40%。（20 分）（实例位置：光盘\mr\12\qjyy\05_diy）

12.6.6　情景应用 6——制作生活画册

视频讲解：光盘\mr\12\lx\制作生活画册.exe

实例位置：光盘\mr\12\qjyy\06

上传图片是一个比较常见的功能，但是在输出上传图片时可能会遇到一些问题，由于上传图片的大小没有固定，从而可能会导致图片在输出时变形，这是一个很让人伤脑筋的问题。如果在上传图片时直接将其生成一个固定大小的缩略图，并同时保存上传的原始图片，那么在输出时就不会有任何问题。在本实例中就实现了这样一个功能，在将图片上传到服务器的同时，生成图片的缩略图，直接浏览图片的缩略图，而如果要下载，则下载的仍是原始图片。本实例的运行结果如图 12.19 所示。





图 12.19　缩略图艺术库

操作步骤如下：

（1）在上传文件的处理页 index_ok.php 中，包含了一个 thumbnail.php 文件；重新定义上传文件的类型，并重新指定上传文件的存储位置，定义原始文件存储文件夹和缩略图存储文件夹；在通过 move_uploaded_file()函数完成文件的上传操作后，调用 thumbnail.php 文件中包含的方法 makethumb()将上传的图片生成缩略图，并添加文字水印。这是在 index_ok.php 文件中所作的处理，关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include "conn/conn.php";
include "thumbnail.php";
if ($_POST [files] != "") {
    $data = date ( "Y-m-d H:m:s" );
    function check($var) {                                              //验证数组的返回值是否为空
        return ($var != "");
    }
    $files = array_filter ( $_POST ["files"], "check" );                //去除数组中空值
    $array = array_filter ( $_FILES ["picture"] ["name"], "check" );    //去除数组中空值
    foreach ( $array as $key => $value ) {                              //循环读取数组中数据
        $types = strtolower ( strstr ( $value, '.' ) );                 //截取上传文件的后缀
        if ($types == ".jpg" || $types == ".gif" || $types == ".png") { //判断上传文件的后缀是否符合要求
            $thumbnail = 'thumbnail/';
            $original = 'original/';
            $file_name = time () . $key . strtolower ( strstr ( $value, "." ) ); //定义上传文件名称
            $path = $original . $file_name;                             //定义原始文件的存储位置
            $path_thumbnail = $thumbnail . $file_name;                  //定义缩略图的存储位置
            move_uploaded_file ( $_FILES ["picture"] ["tmp_name"] [$key], $path );   //执行上传操作
            makethumb ( $path, $path_thumbnail, "200", "200", "100", "www.mrbccd.com" );   //生成缩略图
            $query = "insert into tb_up_file (file_test,path_thumbnail,data,file_name) values
('$path','$path_thumbnail','$data','$files[$key]')";
            $result = mysql_query ( $query );
        } else {
            echo "上传文件" . $value . "类型不正确！";
        }
    }
```

```
        echo "<script>alert('图片上传成功'); window.location.href='index.html';</script>";
}
?>
```

（2）编写 thumbnail.php 文件，定义 makethumb()方法，完成缩略图的生成和水印的添加操作。自定义方法 makethumb()的语法如下：

```
makethumb($srcFile, $dstFile, $dstW, $dstH, $rate = 100, $markwords = null, $markimage = null){}
```

参数说明：

☑ $srcFile：图片文件名。
☑ $dstFile：另存文件名。
☑ $markwords：水印文字。
☑ $markimage：水印图片。
☑ $dstW：图片保存宽度。
☑ $dstH：图片保存高度。
☑ $rate：图片保存品质，默认值为 100。

DIY：将图 12.19 的图片水印内容更改为明日科技。（20 分）（实例位置：光盘\mr\12\qjyy\06_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	6	总分数	
分数								

12.7 自我测试

一、选择题（每题 10 分，5 道题）

1．Jpgraph 是用（　　）编程语言编写的。

A．C　　B．Java　　C．PHP

2．如何判断 PHP 是否支持 GD2 函数库（　　）。

A．连接数据库　　B．浏览 phpinfo.php 页面

C．浏览 php.ini 文件

3．rand()函数的主要用途是（　　）。

A．获取随机数　　B．转换字符串　　C．加密操作

4．告知浏览器页面输出的是一个图像需要使用函数（　　）。

A．rand　　B．microtime　　C．header

5．看下面一行代码：

```
Header("Content-Type:image/jpeg");
```

此代码告知浏览器输出图像的格式为（　　）。

A．gif　　B．png　　C．jpeg

二、填空题（每题 10 分，5 道题）

1．请将下面的代码补充完整，输出 800*800 的图像。



Header("Content-Type:image/jpeg");
（　　　　　　　　　　　　　　　　　　）
Imagejpeg($image);

2．销毁图像需要使用（　　　　　）函数。

3．创建画布需要使用（　　　　　）函数。

4．GD2 函数中绘制中文字需要使用（　　　　　）函数。

5．程序运行后，显示的图片长宽分别为（　　）和（　　）。

```
<?php
    header("Content-Type:image/jpeg");
    $image=imagecreate(130,30);
    $bgcolor=imagecolorallocate($image,230,230,230);
    $fontcolor=imagecolorallocate($image,70,70,70);
    imagestring($image,5,16,6,"A B C D E F",$fontcolor);
    imagejpeg($image);
    imagedestroy($image);
?>
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

12.8　行 动 指 南

开始日期：______年____月____日　　　　　　　　　　　　　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目 分数（　　）	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
		分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目 分数（　　）	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
		分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目 分数（　　）	分数>75 分	优秀，有成为编程高手的潜质。
		分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）制作企业销售柱形图。	
		（2）制作企业销售多柱形图。	
		（3）制作彩票走势折线图。	
		（4）制作食物营养饼形图。	

续表

序号	内　　容	行 动 指 南
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

12.9　成功可以复制——因特网的点火人马克·安德森

成就： 1995 年 8 月 9 日的网景公司首次公开发行股票（IPO）标志着“点 COM”时代的到来。网景领航员（Netscape）浏览器开发者。

创始公司： 网景（Netscape）公司创始人

财富巅峰： 24 岁时，马克·安德森凭借网景公司上市，成为年轻的亿万富豪。

最高学历： 本科

超级语录： 在互联网的光芒下，操作系统只不过是一套“漏洞重重的设备驱动器”。

成功历程

安德森出生在威斯康星州一个小镇的普通家庭，9 岁开始接触计算机，在图书馆自学 Basic 语言。当升入六年级时，安德森已经创造出一个替他做数学家庭作业的虚拟计算器。七年级时，他编写出一个在自家计算机上玩的游戏程序。1993 年，刚刚大学毕业的安德森还没有找到更好的工作，于是就和几个志同道合的朋友一起写 Internet 浏览软件。早期的万维网简陋，只有文本，没有图像、声音、色彩，只是一个基于文本之上，仅仅通过原始界面才能进入的网络。安德森在使用万维网颇感不便的同时，敏锐地察觉到开发一个带有图形、易于客户使用的网页浏览器的潜在市场。经过不断探索和坚持，1994 年 4 月，安德森带领他的工作组开发成功 Navigator（领航员）浏览器，随后该浏览器的销售在 Internet 上如“凯歌般地行进”，一下就占据了 80%以上的份额。



1995 年 8 月 9 日，互联网时代到来了，成立还不到 16 个月、从未赢利过的 Netscape 公司在纽约上市。投资银行事先估计每股仅能卖 14 美元左右，然而开盘后股价一路飙升，最高时竟达到让全美纪经人都目瞪口呆的 71 美元，两个小时内 500 万股被抢购一空，这家创始资金只有 400 万美元的小公司一夜之间便成为 20 亿美元的巨人。华尔街日报评论说，通用动力公司花了 43 年才使市值达到 27 亿美元，而网景只花了 1 分钟。年仅 24 岁的安德森也仿佛神话般地从一文不名到拥有 1.74 亿美元身价的“Internet 富翁”。1997 年 7 月的美国《旗帜》周刊更把安德森们称为“无限制资本家”，他们正从根本上重塑着美国社会，推动着从工业经济向信息经济的过渡。



Navigator 1.0 浏览器

深度评价

马克·安德森的成功启示我们，作为编程开发人员，要善于从生活、工作中发现机会，并坚持把一个小机会做到完美，就可以成就大事业。编程知识不在于多，而在于把学到的知识应用于实践，改善我们工作、学习中的低效率或不便。

第13堂课

文件目录处理技术

（视频讲解：95分钟）

掌握文件处理技术对于Web开发者来说是十分必要的。虽然在处理信息方面，使用数据库是多数情况下的选择，但对于少量的数据，利用文件来存取是非常方便快捷的，更关键的是PHP中提供了非常简单方便的文件、目录处理方法。本堂课将对PHP操作文件和目录的技术进行详细讲解。

学习摘要：

- **基本的文件处理技术**
- **目录操作技术**
- **文件的上传技术**

13.1　基本的文件处理技术

文件操作是通过 PHP 内置的文件系统函数完成的。在学习文件和目录操作技术时，有很多读者认为文件系统的内容非常难学，总是让人晕头转向的，不知道该使用哪个函数，哪个函数在前，哪个函数在后。

之所以会出现这种情况，是没有找到学习的方法，没有理清学习的思路。其实学习文件系统可以归纳为如下 3 个步骤：

（1）学习打开文件的方法。

（2）学习读取、写入和操作文件的方法。

（3）学习关闭文件的方法。

把握住这 3 个步骤，就可以区分出函数使用的先后顺序和功能，达到运用自如。

13.1.1　打开一个文件

要对文件进行操作，首先必须要打开这个文件。

在 PHP 中使用 fopen()函数打开一个文件，语法如下：

```
resource fopen (string filename, string mode [, int use_include_path [, resource zcontext]])
```

参数 filename 指定打开的文件名。

> **说明：** 参数 filename 可以是包含文件路径的文件名（如“C:/windows/php.ini”或者“./php.ini”）。为了避免不同系统之间切换可能带来的麻烦，采用“/”作为路径分隔符。参数 filename 也可以是由某种协议给出的 URL（如“http://mrbccd.cn”或者“ftp://www.mrbccd.cn//”）。如果指定 URL 地址，则可以打开远程文件。

参数 mode 设置打开文件的方式，参数值如表 13.1 所示。

表 13.1　fopen()函数中参数 mode 的可选值

mode	模式名称	说　　明
r	只读	读模式——读文件，文件指针位于文件头部
r+	只读	读写模式——读、写文件，文件指针位于文件头部。注意：如果在现有文件内容的末尾之前进行写入就会覆盖原有内容
w	只写	写模式——写入文件，文件指针指向文件头部。注意：如果文件存在，则文件内容被删除，重新写入；如果文件不存在，则函数自行创建文件
w+	只写	写模式——读、写文件，文件指针指向文件头部。注意：如果文件存在，则文件内容被删除，重新写入；如果文件不存在，则函数自行创建文件
x	谨慎写	写模式打开文件，从文件头部开始写入。注意：如果文件已经存在，函数返回 FALSE，产生一个 E_WARNING 级别的错误信息
x+	谨慎写	读/写模式打开文件，从文件头部开始写入。注意：如果文件已经存在，函数返回 FALSE，产生一个 E_WARNING 级别的错误信息
a	追加	追加模式打开文件，文件指针指向文件尾部。注意：如果文件已有内容，则将从文件末尾开始追加；如果文件不存在，则函数自行创建文件

续表

mode	模式名称	说　明
a+	追加	追加模式打开文件，文件指针指向文件尾部。注意：如果文件已有内容，则从文件末尾开始追加或者读取；如果文件不存在，则函数自行创建文件
b	二进制	二进制模式——用于与其他模式进行连接。注意：如果文件系统能够区分二进制文件和文本文件，可能会使用它。Windows 可以区分，UNIX 则不区分。推荐使用这个选项，便于获得最大程度的可移植性。它是默认模式
t	文本	用于与其他模式的结合。这个模式只是 Windows 下的一个选项

参数 use_include_path 为可选参数，决定是否在 include_path（php.ini 中的 include_path 选项）定义的目录中搜索 filename 文件。例如，在 php.ini 文件中设置 include_path 选项的值为“E:\AppServ\www\MR\Instance\10\”，如果希望服务器在这个路径下打开所指定的文件，则设置参数 use_include_path 的值为 1 或 TRUE。

参数 context 称为上下文，同样为可选参数，是设置流操作的特定选项，用于控制流的操作特性。一般情况下只需使用默认的流操作设置，不需要使用此参数。

注意： 这就是 fopen()函数，在使用这个函数时应该谨慎从事，因为从 fopen()函数的参数中可以看到，使用这个函数不像平时用的 note、word 那么简单，一不小心就有可能将文件内容全部删除。

例 13.01　下面编写一个实例，体会一下这个函数的功能。利用 fopen()函数打开指定的文件。**（实例位置：光盘\mr\13\sl\13.01）**

创建 index.php 文件，应用 fopen()函数打开指定文件夹下的文件。首先，创建一个自定义函数 path，将 fopen()函数封装到自定义函数中，传递两个参数值，一个是指定打开的文件，另一个是文件的读取方式，最后返回文件打开后的标识。代码如下：

```
function path($str,$type){                    //自定义函数将 fopen()函数封装到方法中
    return fopen($str,$type);
}
```

然后，创建自定义函数 info()，根据传递的参数值判断文件是否打开成功，并输出提示信息。自定义函数的语法如下：

```
function info($str){                          //自定义函数判断文件打开是否成功并输出提示
    if($str){
        echo "<script>alert('打开文件成功');</script>";
    }else{
        echo "<script>alert('打开文件失败');</script>";
    }
}
```

最后，定义指定访问文件的位置和背景图像的存储位置，调用自定义函数，打开指定的文件，并且判断文件是否打开。代码如下：

```
$path="txt/in.txt";                           //将文件路径保存在变量中
$path1="images/color.bmp";
info(path($path,"r"));                        //调用自定义方法
info(path($path1,"wb"))
```

运行本实例会在页面中输出信息提示，判断文件打开是否成功。本实例由于笔者自定义创建文件夹所以并没有使用“w+”方式打开文件。采用“w+”方式打开文件时十分特殊，如果文件不存在则创建文件；如果文件存在，且本身存在内容则清空文件中所有内容，重新写入。本实例的运行结果如图 13.1 所示。



图 13.1　文件打开后的提示信息

13.1.2　读取文件内容

文件打开之后，就可以进行读取和写入操作了，这里先讲解文件的读取。可以将 PHP 提供的文件读取函数分为 4 类，即读取一个字符、读取一行字符、读取任意长度的字串和读取整个文件。

1．fgetc()函数，读取一个字符

fgetc()函数从文件指针指定的位置读取一个字符。该函数的语法如下：

```
string fgetc ( resource handle )
```

该函数返回一个字符，该字符从 handle 指向的文件中得到，遇到 EOF 则返回 FALSE。

例 13.02　应用 fgetc()函数循环读取 in.txt 文件中的字符。**（实例位置：光盘\mr\13\sl\13.02）**

（1）在当前目录下创建 txt 文件夹，在此文件夹中新建文本文件，命名为 in.txt 文件。双击打开此文件输入如下文本内容，然后保存并关闭。

```
<font size="13" color="red">hello world</font>
```

（2）创建脚本文件命名为 index.php。首先定义文件路径，然后用只读方式打开文件，由于 fgetc()函数只能读取单个字符，所以为了拼接 for 循环的循环条件，这里使用了 filesize()函数判断文本文件的数据长度。最后利用 fgets()函数输出文本数据。

（3）本实例的核心代码如下：

```
<?php
	$path="txt/in.txt";                    //定义文件路径
	$fopen=fopen($path,"r");               //打开指定文件（只读方式）
	$size=filesize($path);                 //获取文件的数据长度
	for($a=0;$a<$size;$a++){               //for 循环语句
		echo fgetc($fopen);                //输出数据
	}
	fclose($fopen);                        //关闭文件
?>
```

运行结果如图 13.2 所示。



图 13.2　循环读取文件数据

2．fgets()函数，读取一行字符

fgets()函数从文件指针中读取一行数据。文件指针必须是有效的，并且必须指向一个由 fopen()或 fsockopen()成功打开的文件。fgets()函数的语法如下：



```
string fgets( int handle [, int length] )
```

其中，参数 handle 是被打开的文件，参数 length 是要读取的数据长度。

fgets()函数能够从 handle 指定文件中读取一行并返回长度最大值为 length-1 个字节的字符串。在遇到换行符、EOF 或者读取了 length-1 个字节后停止，如果忽略 length 参数，那么将读取到行结束。

例 13.03　本实例利用 fgets()和 fgetss()两个函数分别输出文本文件的内容以显示两者的区别。**（实例位置：光盘\mr\13\sl\13.03）**

具体操作步骤如下：

创建 index.php 文件，分别应用 fgets()和 fgetss()函数对指定文本文件 in.txt 进行读取，看输出结果有什么不同。代码如下：

```
<?php
	$path="txt/in.txt";                                          //定义文件路径
	$fopen=fopen($path,"r");                                     //以只读方式打开文件
	echo '利用 fgets 函数输出数据：'.fgetc($fopen);               //fgets 方式读取
	echo "<br>";
	echo '利用 fgetss 函数输出数据：'.fgets($fopen);              //fgetss 方式读取
?>
```

运行结果如图 13.3 所示。



图 13.3　比较 fgets()和 fgetss()函数的输出结果

从图 13.3 中可以看出，应用 fgets()函数读取的数据原样输出，没有任何变化；而应用 fgetss()函数读取的数据，去除了文件中的 html 标记，输出的完全是普通字符串。

说明： fgets()和 fgetss()函数输出数据时需使用各自的 fopen()函数打开文件，否则会出现意想不到的结果。例如，首先使用 fgets()函数读取一行数据，然后利用 fgetss()函数读取同一打开文件的数据时，文件指针自动换到下一行即接着 fgets()函数的下一行进行操作数据。

3．fread()函数，读取任意长度的字串

fread()函数从文件中读取任意长度的数据，还可以用于读取二进制文件。该函数的语法如下：

```
string fread ( int handle, int length )
```

其中，参数 handle 为指向的文件资源，参数 length 指定要读取的字节数。此函数在读取到 length 个字节或者到达 EOF 时停止执行。

例 13.04　应用 fread()函数读取 txt 文件夹下 in.txt 文件的内容。**（实例位置：光盘\mr\13\sl\13.04）**

首先，定义文本文件在实例根目录下的存储位置。其次，利用 fopen()函数以只读方式打开文件并返回文件句柄。然后，利用 filesize()函数获取文本文件数据的长度。最后，利用 fread()函数输出文本数据。代码如下：

```
<?php
	$path="txt/in.txt";				//定义文本文件路径
	$open=fopen($path,"r");			//打开指定文件
	$size=filesize($path);			//获取文本数据长度
	echo fread($open,$size);		//输出所有数据
	fclose($open);					//关闭文件
?>
```

运行结果如图 13.4 所示。

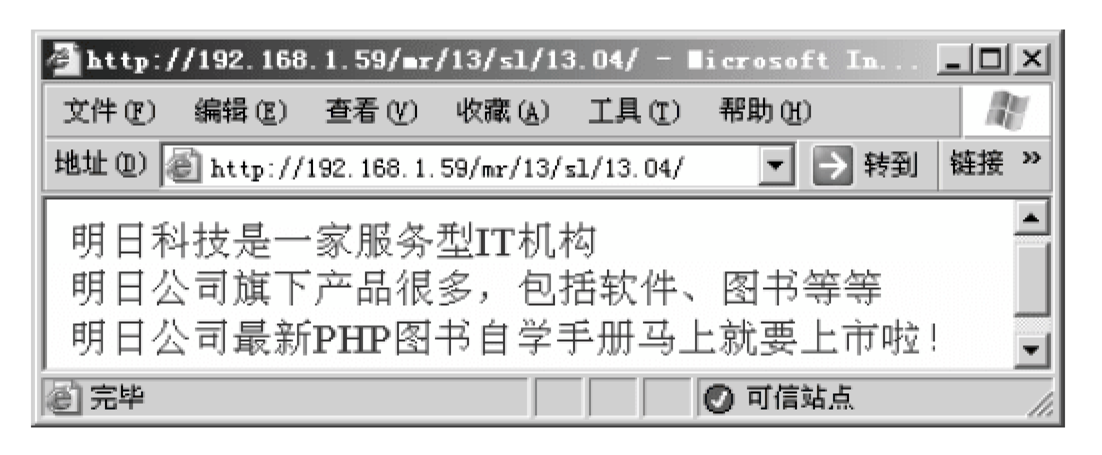


图 13.4　fread()函数的使用

4．readfile()、file()和 file_get_contents()函数，读取整个文件

☑　readfile()函数

readfile()函数读取一个文件并写入到输出缓冲，成功返回读取的字节数，失败返回 FALSE。该函数的语法如下：

```
int readfile ( string filename [, bool use_include_path [, resource context]] )
```

其中，参数 filename 指定读取的文件名称；参数 use_include_path 控制是否支持在 include_path 中搜索文件，如果支持，则将该值设置为 TRUE；参数 context 是 PHP 5.0 新增内容。

注意：应用 readfile()函数，不需要打开/关闭文件，不需要输出语句，直接应用函数即可。

☑　file()函数

file()函数将整个文件的内容读入到一个数组中。成功返回数组，数组中的每个元素都是文件中对应的一行，包括换行符在内；失败返回 FALSE。该函数的语法如下：

```
array file ( string filename [, int use_include_path [, resource context]] )
```

其参数与 readfile()函数相同，唯一区别是该函数的返回值是数组。

☑　file_get_contents()函数

file_get_contents()函数将文件内容读入一个字符串。如果有 offset 和 maxlen 参数，将在参数 offset 所指定的位置开始读取长度为 maxlen 的内容。如果失败，返回 FALSE。该函数的语法如下：

```
string file_get_contents ( string filename [, bool use_include_path [, resource context [, int offset [, int maxlen]]]] )
```

其中，参数 filename 指定读取的文件名称；参数 use_include_path 控制是否支持在 include_path 中搜索文件，如果支持，则将该值设置为 TRUE。

例 13.05　在了解了这些函数之后，下面编写一个实例，体会一下这些函数的功能。读取指定文件中的全部内容。（**实例位置：光盘\mr\13\sl\13.05**）

创建 index.php 文件，分别应用 readfile()、file()和 file_get_contents()函数读取指定文件中的内容。代码如下：

```
<?php
function type($number,$path="txt/in.txt"){				//自定义函数，将 path 参数定义为可选参数
```



```
        if($number=="1"){                                    //判断传递进来的参数是否等于 1
            echo '<h2>file_get_contents()输出数据</h2>';      //输出数据
            echo file_get_contents($path);
        }else{
            if($number=="2"){                                //判断传递进来的参数是否等于 2
                echo '<h2>readfile()输出数据</h2>';            //输出数据
                readfile($path);
            }else{
                $array=file($path);                          //将数据保存在数组中
                echo '<h2>file()输出数据</h2>';                //循环输出数据
                for($a=0;$a<count($array);$a++){
                    echo "#".$array[$a]."<br>";
                }
            }
        }
}
type("3");                                                   //方法调用
type("2");
type("1");
?>
```

运行结果如图 13.5 所示。

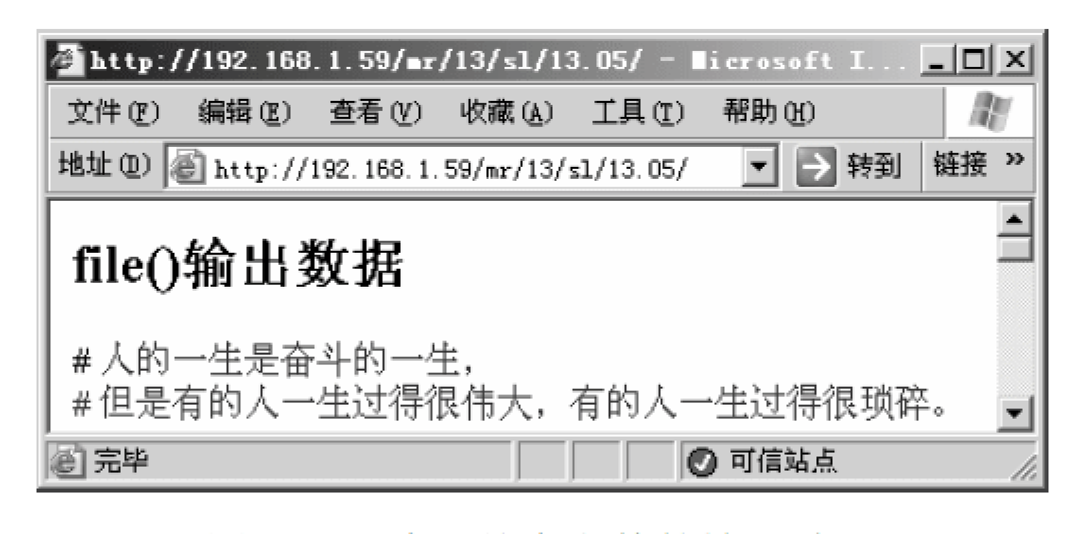


图 13.5　读取整个文件的输出结果

注意：不知道大家是否已经注意到，在通过 readfile()、file()和 file_get_contents()函数读取整个文件中的内容时，不需要通过 fopen()函数打开文件，也不需要使用 fclose()函数关闭文件。但是，在读取一个字符、读取一行字符和读取任意长度的字符串时必须应用 fopen()函数打开文件后才能进行读取，在读取完成后还要应用 fclose()函数关闭文件。

13.1.3　向文件中写入数据

前面讲了打开和读取，下面介绍文件的写入操作。PHP 中通过 fwrite()和 file_put_contents()函数执行文件的写入操作。

1．fwrite()函数，向文件中写入数据

fwrite()函数执行文件的写入操作，它还有一个别名 fputs()。其语法如下：

```
int fwrite ( resource handle, string string [, int length] )
```

fwrite()函数把 string 的内容写入文件指针 handle 处。如果设置 length，那么当写入 length 个字节或者完成 string 的写入后，操作就会停止。fwrite()函数成功返回写入的字符数，失败则返回 FALSE。

注意：在应用 fwrite()函数时，如果给出 length 参数，那么 magic_quotes_runtime（php.ini 文件中的选项）配置选项将被忽略，而 string 中的斜线将不会被抽去。如果在区分二进制文件和文本文件的系统上（如 Windows）应用这个函数，打开文件时，fopen()函数的 mode 参数要加上'b'。

2. file_put_contents()函数，向文件中写入数据

file_put_contents()函数将一个字符串写入文件中。成功返回写入的字节数，失败则返回 FALSE。其语法如下：

```
int file_put_contents ( string filename, string data [, int flags [, resource context]] )
```

file_put_contents()函数的参数说明如表 13.2 所示。

表 13.2　file_put_contents()函数的参数说明

参　　数	说　　明
filename	指定写入文件的名称
data	指定写入的数据
flags	实现对文件的锁定。可选值为 FILE_USE_INCLUDE_PATH、FILE_APPEND 或 LOCK_EX，这里只要知道 LOCK_EX 的含义即可，LOCK_EX 为独占锁定
context	一个 context 资源

提示：本函数可安全用于二进制对象。如果“fopen wrappers”已经被激活，则在本函数中可以把 URL 作为文件名来使用。

例 13.06　在了解了这两个函数的功能之后，下面编写一个实例，通过实例来体会一下它们的功能和不同之处。**（实例位置：光盘\mr\13\sl\13.06）**

通过 fwrite()和 file_put_contents()函数执行文件的写入操作。具体步骤如下：

创建 index.php 文件，其中包括两个 PHP 脚本文件。第一个应用 file_put_contents()函数写入文件。首先应用 file_get_contents()函数读取 bg_01.jpg 图片文件，然后将读取到的二进制数据通过 file_put_contents()函数写入到另外一个 files.jpg 文件中，最后通过 img 标记输出 files.jpg 图片。代码如下：

```
<?php
	$path="images/work_03.gif";                              //图片地址和名称
	$pic=file_get_contents($path);                           //获取数据信息并保存到变量中
		file_put_contents("images/work_04.gif",$pic);        //将图片信息写入到另一张图片中
	echo "<img src='images/work_04.gif'>";                   //显示图片
?>
```

第二个应用 fwrite()函数完成文件的写入操作。代码如下：

```
<?php
	$path="images/work_03.gif";                              //图片地址和名称
	$pic=file_get_contents($path);                           //读取图片数据
	$open=fopen("images/work_04.gif","wb");                  //以读写二进制方式打开文件
	fwrite($open,$pic);                                      //写入信息
	echo "<img src='images/work_04.gif'>";                   //输出图像
	fclose($open);                                           //关闭文件
?>
```

运行结果如图 13.6 所示。

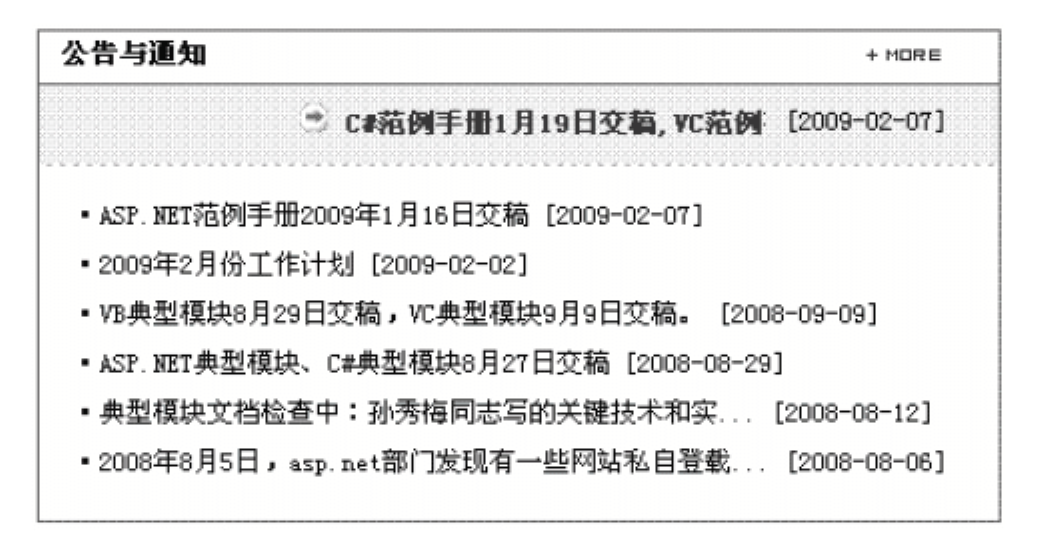


图 13.6　输出写入的二进制文件

13.1.4　关闭文件指针

文件有打开就应该有关闭，对文件的操作结束后，应该关闭这个文件，否则可能引起错误。在 PHP 中使用 fclose()函数关闭文件。其语法如下：

```
bool fclose ( resource handle ) ;
```

fclose()函数将参数 handle 指向的文件关闭，成功返回 TRUE，否则返回 FALSE。其中参数 handle（文件指针）必须是有效值，并且是通过 fopen()函数成功打开的文件。有关该函数的应用可以参考上面的实例，这里不再赘述。

13.2　目录操作技术

目录也是文件，是一种特殊的文件。那么既然是文件，如果对其进行操作同样必须要先打开，然后才可以进行浏览、操作，最后还要记得关闭。对于 PHP 目录处理技术的学习仍然可以依照学习文件操作技术的步骤进行。

13.2.1　打开指定目录

打开文件和打开目录虽然都是执行打开的操作，不但使用的函数不同，而且对未找到指定文件的处理结果也不同。fopen()函数如果未找到指定的文件，那么可能会自动创建这个文件，而打开目录函数 opendir()却没有那么勤劳和爽快，它会直接抛出一个错误信息。这就是 PHP 提供的打开目录的函数 opendir()。

opendir()函数打开一个指定目录。成功则返回目录句柄，否则返回 FALSE。其语法如下：

```
resource opendir ( string path [, resource context] )
```

其中，参数 path 指定要打开的目录路径，如果参数 path 指定的不是一个有效的目录，或者因为权限、文件系统错误而不能打开，opendir()函数将返回 FALSE，并产生一个 E_WARNING 级别的错误信息。

提示：通过在 opendir()函数前添加@符号，可以屏蔽错误信息的输出。

13.2.2　读取目录结构

应用 opendir()函数打开目录之后，就可以利用其返回的目录句柄，配合 PHP 中提供的 scandir()函数完成对目录的浏览操作。

scandir()函数浏览指定路径下的目录和文件。成功返回包含有文件名的 array，失败则返回 FALSE。其

语法如下：

```
array scandir ( string directory [, int sorting_order [, resource context]] )
```

其中，参数 directory 指定要浏览的目录，如果 directory 不是目录，那么 scandir()函数将返回 FALSE，并生成一条 E_WARNING 级的错误信息；参数 sorting_order 设置排序顺序，默认按字母升序排序，如果应用参数 sorting_order，则变为降序排序。

例 13.07 下面编写一个实例，体会一下打开目录和浏览目录的功能。本实例的主要用意是浏览 Apache 服务器的根目录。**（实例位置：光盘\mr\13\sl\13.07）**

创建 index.php 文件，首先定义 Apache 根目录的相对路径，利用 realpath()函数输出根目录的绝对路径。然后，利用 scandir()函数将目录下的信息保存在数组中，最后利用循环语句将数组信息循环输出。代码如下：

```
<?php
	$path="../../../../";                                            //定义相对路径
	echo "Apache 根目录所在的硬盘路径为：".realpath($path)."<br>";     //输出 Apache 根目录的绝对路径
	if(is_dir($path)){                                               //判断当前路径是否为目录
		$path=scandir($path);                                        //将目录信息保存在数组中
		for($a=0;$a<count($path);$a++){                              //循环输出结果
			echo "#".$path[$a]."<br>";
		}
	}
?>
```

运行结果如图 13.7 所示。

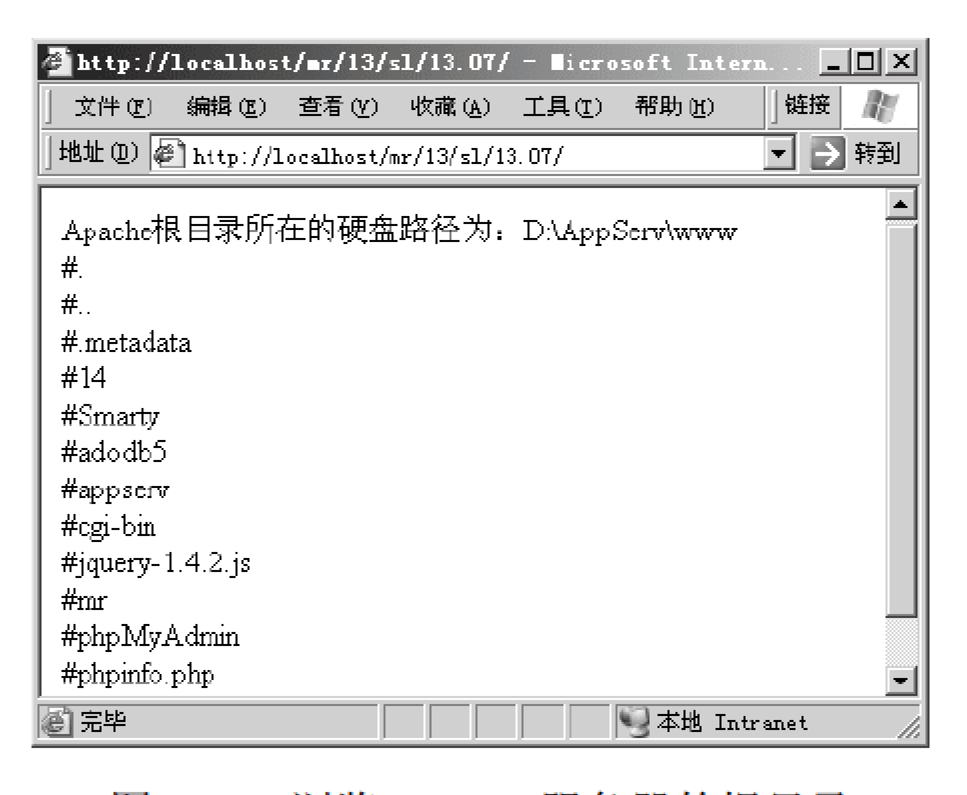


图 13.7 浏览 Apache 服务器的根目录

13.2.3 关闭目录指针

人做事情应该有始有终，编程也应该是如此。打开目录，完成操作之后，就应该关闭目录。PHP 中通过 closedir()函数关闭目录。其语法如下：

```
void closedir ( resource handle )
```

其中，参数 handle 为使用 opendir()函数打开的一个目录句柄。

13.3 文件的上传技术

学习文件上传技术，其归纳的步骤如下：第一步是学习如何控制上传文件（配置 php.ini 文件），第二步



是学习对上传文件判断（判断上传文件的格式、大小），第三步是学习执行文件上传操作的方法。

13.3.1　开启文件上传功能

PHP 中通过 php.ini 文件对上传文件进行控制，包括是否支持上传、上传文件的临时目录、上传文件的大小、指令执行的时间、指令分配的内存空间。

在 php.ini 文件中，定位到 File Uploads 项，完成对上传相关选项的设置。上传相关选项的含义介绍如下。

- ☑ file_uploads：如果值是 On，说明服务器支持文件上传；如果为 Off，则不支持。一般默认是支持的，这个不用修改。
- ☑ upload_tmp_dir：上传文件临时目录。在文件被成功上传之前，文件首先存放到服务器端的临时目录中。多数使用系统默认目录，但是也可以自行设置。
- ☑ upload_max_filesize：服务器允许上传文件的最大值，以 MB 为单位。系统默认为 2MB，如果网站需要上传超过 2MB 的数据，那么就要修改这个值。

上述是 php.ini 文件中 File_Uploads 项中与上传相关选项参数设置说明，除了 File_Uploads 项中的内容外，在 php.ini 文件中还有其他几个选项会影响到文件的上传。

- ☑ max_execution_time：PHP 中一个指令所能执行的最大时间，单位是秒。该选项在上传超大文件时必须要修改，否则即使上传文件在服务器允许的范围内，但是超过了指令所能执行的最大时间，仍然无法实现上传。
- ☑ memory_limit：PHP 中一个指令所分配的内存空间。单位是 MB。它的大小同样会影响到超大文件的上传。

注意：（1）php.ini 文件配置完成后，需要重新启动 Apache 服务器，配置才能生效。（2）如果使用集成化的安装包来配置 PHP 的开发环境，那么就不必担心上述介绍的这些配置信息，因为默认已经配置好。（3）但是，如果要上传超大的文件，那么就有必要对 php.ini 进行修改（即使使用集成化安装包配置的也不行）。修改选项包括 upload_max_filesize 的最大值、max_execution_time（一个指令所能执行的最大时间）和 memory_limit（一个指令所分配的内存空间）。

13.3.2　全局变量$_FILES 应用

对上传文件进行判断应用的是全局变量$_FILES，$_FILES 是一个数组，它包含所有上传的文件信息。下面介绍$_FILES 数组中每个元素的含义，如表 13.3 所示。

表 13.3　$_FILES 数组中每个元素的含义

元　素　名	说　　明
$_FILES[filename][name]	存储上传文件的文件名，如 text.txt、title.jpg 等
$_FILES[filename][size]	存储文件大小。单位为字节
$_FILES[filename][tmp_name]	存储文件在临时目录中使用的文件名。因为文件在上传时，首先要将其以临时文件的身份保存在临时目录中
$_FILES[filename][type]	存储上传文件的 MIME 类型，MIME 类型规定各种文件格式的类型。每种 MIME 类型都是由“/”分隔的主类型和子类型组成。如“image/gif”，主类型为“图像”，子类型为 GIF 格式的文件，“text/html”代表文本的 HTML 文件

续表

元 素 名	说 明
$_FILES[filename][error]	存储与文件上传相关的错误代码。此项目是 PHP 4.2.0 版本中新增内容。其返回值有 5 种，如下所示。 ☑ @0：表示没有任何错误，文件上传成功 ☑ @1：表示上传文件的大小超出了 PHP 配置文件指令 upload_max_filesize 选项限制的值 ☑ @2：表示上传文件大小超出了 HTML 表单中 MAX_FILE_SIZE 选项所指定的值 ☑ @3：表示文件只被上传了一部分 ☑ @4：表示没有上载任何文件

在 $_FILES 数组元素中，最为常用的是 $_FILES[filename][name]、$_FILES[filename][size] 和 $_FILES[filename][tmp_name]。通过这 3 个元素值完成上传文件名称的获取，判断上传文件的格式（根据上传文件名称的后缀），判断上传文件的大小。

例 13.08 下面编写一个实例，应用$_FILES 获取上传文件名称、上传文件的格式和上传文件大小等。**（实例位置：光盘\mr\13\sl\13.08）**

创建 index.php 文件，向页面中添加表单，通过使用 POST 方法提交数据，设置 enctype 属性为"multipart/form-data"，设置 action 属性为 index.php。添加隐藏域、文件域和提交按钮，最后通过$_FILES 全局数组获取上传文件的相关信息。代码如下：

```
<?php
if($_SESSION['time']==$_POST[time]){                                    //判断页面是否刷新
        if(isset($_POST[sub])){                                         //判断页面中是否存在 sub 变量
                $file=$_FILES[name];                                    //将文件信息保存在数组中
                echo "上传文件的路径及名称：".$file['tmp_name']."<br>";  //输出上传文件的路径名称
                echo "上传文件的名称：".$file['name']."<br>";            //输出上传文件的名称
                echo "上传文件的类型：".$file['type']."<br>";            //输出上传文件的类型
                echo "上传文件的错误信息：".$file['error']."<br>";       //输出可能发生的错误信息
                echo "上传文件的大小：".$file['size'];                   //输出文件大小
        }
}
$time=mt_rand(0,10000);                                                 //获取一个随机数
$_SESSION['time']=$time;                                                //将随机数保存在变量中
  ?>
```

运行结果如图 13.8 所示。



图 13.8　获取上传文件的信息



13.3.3　将上传文件移动到指定目录

PHP 中应用 move_uploaded_file()函数实现文件上传。但是，在执行文件上传之前，为了防止潜在的攻击对原本不能通过脚本交互的文件进行非法管理，可以先应用 is_uploaded_file()函数判断指定的文件是否是通过 HTTP POST 上传的，如果是则返回 TRUE，则可以继续执行文件的上传操作，否则将不能够继续执行。

1. is_uploaded_file()函数

is_uploaded_file()函数判断指定的文件是否是通过 HTTP POST 上传的。其语法如下：

```
bool is_uploaded_file ( string filename )
```

其中，参数 filename 必须指定类似于$_FILES['filename']['tmp_name']的变量，不可以使用从客户端上传的文件名$_FILES['filename']['name']。

通过 is_uploaded_file()函数对上传文件进行判断，可以确保恶意的用户无法欺骗脚本去访问本不能访问的文件，如/etc/passwd。

2. move_uploaded_file()函数

move_uploaded_file()函数将文件上传到服务器中指定的位置。如果成功返回 TRUE，否则返回 FALSE。其语法如下：

```
bool move_uploaded_file ( string filename, string destination )
```

其中，参数 filename 指定上传文件的临时文件名，即$_FILES[tmp_name]；参数 destination 指文件上传后保存的新路径和名称。

说明： 如果参数 filename 不是合法的上传文件，不会执行任何操作，move_uploaded_file()将返回 FALSE。如果参数 filename 是合法的上传文件，但出于某些原因无法移动，同样也不会执行任何操作，move_uploaded_file()将返回 FALSE。此外还会发出一条警告。

例 13.09　下面编写一个实例，应用 move_upload_file()函数实现文件的上传。（**实例位置：光盘\mr\13\sl\13.09**）

在本实例中，通过 is_uploaded_file()函数判断指定的文件是否是通过 HTTP POST 上传，并运行 move_upload_file()函数将上传文件移动到指定的文件夹中。具体步骤如下：

首先，创建表单，设置 enctype 的属性值为 multipart/form-data，设置隐藏域、文件域和提交按钮，使用 POST 方法将数据提交到当前页面。然后，通过$_FILES 获取上传文件的相关信息，并根据上传信息将文件保存在服务器的临时文件夹中。最后，通过 move_upload_file()函数将上传的文件移动到指定位置。代码如下：

```
<?php
	if(isset($_POST[sub])){                          //当页面中存在变量 sub 时
		if(!is_dir("images")){                      //如果不存在 images 文件夹
			mkdir("images");                        //创建此文件夹
		}
		$file=$_FILES['text'];                      //将图片的相关信息保存在数组中
		if(is_uploaded_file($file['tmp_name'])){    //判断文件是否是一个上传文件
			$rand=rand(1,1000);                     //随机取 1～1000 的随机数一个
			$floatTime=microtime();                 //获取此时刻的微秒数
			$str=substr($file['name'],-4,4);        //截取上传文件名称后 4 位
```

```
                $path="images/".($and+$floatTime).$str;        //拼接图片路径
                if(move_uploaded_file($file['tmp_name'],$path)){//将文件移动到指定文件夹并输出信息
                    echo "上传成功，文件名称为：".($and+$floatTime).$str;
                }
            }
        }
?>
```

运行结果如图 13.9 所示。



图 13.9　上传文件

13.3.4　文件下载技术

有上传就应该有下载，这里介绍通过 HTTP 方式下载文件。其主要应用 header()函数。

header()函数，属于 HTTP 函数。其作用是以 HTTP 协议将 HTML 文档的标头送到浏览器，并告诉浏览器具体怎么处理这个页面。header()函数的语法如下：

```
void header ( string string [, bool replace [, int http_response_code]] )
```

参数说明：

☑ string：发送的标头。

☑ replace：如果一次发送多个标头，对于相似的标头是替换还是添加。如果是 FALSE，则强制发送多个同类型的标头。默认是 TRUE，即替换。

☑ http_response_code：强制 HTTP 响应为指定值。

通过 HTTP 下载的代码如下：

```
header("Content-type: application/x-gzip");
header("Content-Disposition: attachment; filename=文件名");
header("Content-Description: PHP3 Generated Data"); >
```

HTTP 标头有很多，这里介绍的是下载的 HTTP 标头。代码如下：

```
header('Content-Disposition: attachment; filename="filename"');
```

在应用的过程中，唯一需要改动的就是 filename。即将 filename 替换为要下载的文件。

例 13.10　下面通过一个具体的实例，讲解如何运用 header()函数完成文件的下载操作。（**实例位置：光盘\mr\13\sl\13.10**）

具体步骤如下：

（1）通过 Content-Type 指定文件的 MIME 类型。

（2）通过 Content-Disposition 对文件进行描述，值“attachment;filename="test.jpg"”说明是一个附件，同时指定下载文件名称。

（3）通过 Content-Length 设置下载文件的大小。

（4）通过 readfile()函数读取文件内容。

具体代码如下：



```
header('Content-Type:image/jpg');                                  //设置图片类型
header('Content-Disposition:attachment;filename="test.jpg"');      //描述下载文件，指定文件名称
header('Content-Length:'.filesize('test.jpg'));                    //定义下载文件大小
readfile('test.jpg');                                              //读取文件，执行下载
```

运行结果如图 13.10 所示。



图 13.10　下载文件

13.4　照猫画虎——基本功训练

13.4.1　基本功训练 1——通过文本文件统计网站访问量

视频讲解：光盘\mr\13\lx\通过文本文件统计网站访问量.exe

实例位置：光盘\mr\13\zmhh\01

统计网站访问量的实例已经在第 9 堂课中得以实现，下面利用操作文本文件的知识实现这一功能，运行结果如图 13.11 所示。

网站已被访问14次

图 13.11　文本文件统计网站的访问量

操作步骤如下：

（1）创建脚本文件 index.php，新建文本文件 in.txt。在 in.txt 文件中给定一个数字“0”。

（2）启动 SESSION，利用 time()函数获取当前时刻的时间戳并将此时间戳保存在 SESSION 变量中。定义一个隐藏域信息框，将时间戳的值赋给隐藏域的 value 属性中。

（3）如果 SESSION 变量的值与$_POST 方法获取到的隐藏域的 value 属性值相同时，程序向下执行，获取文本文件中存储的数据，并且将数据加 1 后输出。然后将加 1 后的值重新写入文本文件，否则直接输出文本文件中的值。程序代码如下：

```
<?php
        session_start();                                        //启用 SESSION
        if($_SESSION[time]==$_POST[hidden]){                    //判断是否是刷新页面
            $number=file_get_contents("web/in.txt ");           //获取文本文件中的值
            $number=$number+1;                                  //将获取到的值加 1
            echo "网站已被访问<b>".$number."</b>次";            //输出信息
            file_put_contents("web/in",$number);                //写入文件
        }else{
            $number=file_get_contents("web/in.txt ");           //读取文本文件中的信息
            echo "网站已被访问<b>".$number."</b>次";            //输出数据
        }
    ?>
```

```
        <input type="hidden" name="hidden" value="<?php $time;?>"> //隐藏域
    <?php
        $time=time();                                              //当前时间戳
        $_SESSION[time]=$time;                                     //将时间戳保存在 SESSION 中
    ?>
```

照猫画虎：在上述实例基础上，同时输出系统的当前时间，并且字体效果一致。（20 分）（实例位置：光盘\mr\13\zmhh\01_zmhh）

13.4.2　基本功训练 2——读取文本文件中存储的用户注册协议

视频讲解：光盘\mr\13\lx\读取文本文件中存储的用户注册协议.exe

实例位置：光盘\mr\13\zmhh\02

注册协议相信用户并不陌生，它是建立网站的管理者对使用网站的用户发布的简要声明，其中可能要求用户要使用文明用语或保护软件版权等。大部分网站在编写此模块功能时一般都会应用到文件操作的知识。下面将模仿并实现这一功能，运行结果如图 13.12 所示。

图 13.12　注册条款

操作步骤如下：

（1）创建脚本文件 index.php，新建文本文件并命名为 tk.txt，将注册条款信息保存在文件中。

（2）编写文本域标签<textarea>，利用 file()函数将文本文件信息保存在数组中，最后利用 foreach()语句循环输出数组的值。程序运行代码如下：

```
<?php
    $array=file("tk/tk.txt");                    //将文本文件信息保存在数组中
?>
<span>注册条款<br></span>
<textarea name="text" cols="40" rows="6">        //编写文本域标签
<?php
    foreach($array as $key=>$value){             //循环数组元素
        echo $value;
    }
?>
</textarea>
```

照猫画虎：在上述实例基础上添加一个单选按钮，由用户决定是否同意条款，使本实例更加贴近实战。（20 分）（实例位置：光盘\mr\13\zmhh\02_zmhh）

13.4.3　基本功训练 3——通过文本文件存储网站访问记录

视频讲解：光盘\mr\13\lx\通过文本文件存储网站访问记录.exe

实例位置：光盘\mr\13\zmhh\03

网站的访问记录一般被所有程序员所采用，它的主要功能是当程序遭到恶意破坏时查找凶手用的。网站访问记录一般包含用户的 IP 地址、主机名称和访问时间等。下面将实现这一功能，运行结果如图 13.13 所示。

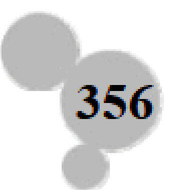




图 13.13　网站访问记录

操作步骤如下：

（1）创建脚本文件 index.php，启动 SESSION 抑制刷新操作。

（2）首先，判断文件夹 Record 是否存在，如果不存在则创建此文件夹。利用 fopen()函数创建并打开文本文件 record.txt，然后利用$_SERVER 函数获取客户机 ip、客户机主机名称并将信息写入文件。最后关闭打开句柄。实现代码如下：

```
<?php
header("Content-Type:text/html;charset=utf-8");
session_start();
if(!is_dir('Record')){                          //判断 Record 文件夹是否存在
    mkdir('Record');                            //创建文件夹
}
if($_SESSION['time']==$_POST['hidden']){        //防刷新操作
    $fopen=fopen("Record/record.txt","a+");     //打开文件（追加方式）
    $ip="IP：".$_SERVER['REMOTE_ADDR'];         //获取客户端 ip
    $host="PC NAME：".$_SERVER['HTTP_X_FORWARDED_FOR'];    //获取客户端主机名称
    $time="time：".date("Y-m-d H:i:s");          //获取当前时间
    $string=$ip."|".$host."|".$time."*||*";      //拼接字符串
    if(is_writable("Record/record.txt")){       //判断文件是否可写
        fwrite($fopen,$string);                 //写入操作
    }
    fclose($fopen);                             //关闭连接
}
?>
<input type="hidden" name="hidden" value="<?php echo $tim; ?>">
<?php
    $tim=time();
    $_SESSION['time']=$tim;
?>
```

照猫画虎：在上面的实例中，再添加一个当前服务器的 IP 地址以便测试哪些操作是由服务器发出的。（20 分）（实例位置：光盘\mr\13\zmhh\03_zmhh）

13.4.4　基本功训练 4——验证指定的目录是否存在

视频讲解：光盘\mr\13\lx\验证指定的目录是否存在.exe

实例位置：光盘\mr\13\zmhh\04

验证指定目录是否存在是非常必要的，因为开发一个任何大小的项目都会应用到不止一个文件夹，如果将信息或数据写入到错误文件夹的文件中，这对于程序员来说都是非常致命的。所以用户无论是否确定某文件夹存在，都要先使用函数进行验证。本实例的运行结果如图 13.14 所示。



图 13.14　判断指定目录是否存在

实现本实例的原理十分简单，主要是利用 is_dir()函数实现验证，语法格式如下：

```
bool is_dir ( string filename )
```

如果文件名存在并且为目录则返回 TRUE。如果 filename 是一个相对路径，则按照当前工作目录检查其相对路径。

程序代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                          //设置编码
    if(is_dir("dir")){                                                        //判断目录是否存在
        echo "<b style='font-family:楷体_gb2312'>指定文件夹存在</b>";         //输出提示信息
    }else{
        echo "<b style='font-family:\'楷体_gb2312\'>指定文件夹不存在</b>";
    }
?>
```

照猫画虎：在验证指定的文件夹存在的同时输出当前文件夹的目录结构。（20 分）（实例位置：光盘\mr\13\zmhh\04_zmhh）

13.4.5　基本功训练 5——控制上传文件的大小和格式

视频讲解：光盘\mr\13\lx\控制上传文件的大小和格式.exe

实例位置：光盘\mr\13\zmhh\05

在网站的开发中，控制上传文件的大小和格式是非常常见的。因为网站服务器的资源很有限，不可能为上传模块占用网站过多资源。另一方面，恶意的程序或软件是网站被黑的重要原因，所以控制上传文件的格式是十分有必要的。为了解决这些问题，笔者制作了控制上传文件的大小和格式模块，运行结果如图 13.15 所示。



图 13.15　控制上传文件的大小和类型

操作步骤如下：

（1）创建脚本文件并命名为 index.php。

（2）编写文件上传域，当“上传”按钮被单击时，获取上传文件的相关信息并保存在数组中。

（3）通过$_FILES[]全局数组获取上传文件的相关信息，对上传文件的信息进行判断，并且输出提示信息。代码如下：

```
<?php
    if($_POST[sub]){                                                  //单击“上传”按钮
        $file=$_FILES['filename'];                                    //将上传文件的相关信息保存在数组中
        if($file['size']>1024*1000 and $file['type']!="image/pjpeg"){ //定义条件
            echo "上传文件不是指定类型或超出指定大小";                  //输出提示
        }else{
```

```
                echo "上传成功<br>";
                echo "上传文件的类型为<b>".$file['type']."</b><br>"; //输出上传文件的格式
                echo "上传文件的大小为<b>".$file['size']."</b>";      //输出上传文件的大小
            }
        }
?>
```

照猫画虎：在上面的实例中，再添加一个文件判断是否通过 POST 方式上传的验证，并达到同样的输出效果。（20 分）（实例位置：光盘\mr\13\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

13.5　情景应用——拓展与实践

13.5.1　情景应用 1——规范化的文件上传

视频讲解：光盘\mr\13\lx\规范化的文件上传.exe

实例位置：光盘\mr\13\qjyy\01

文件上传一直是网站开发者所青睐的模块，它可以使用户与网站进行有效的互动。但是粗略的上传模块制作会使网站的开发者得不偿失，主要表现在如下两点：一是上传文件的名称有可能出现重名的现象，而一般的操作系统不允许文件重名，一旦重名则新上传的文件会覆盖原有文件。二是大文件上传。当上传文件过大时首先会消耗服务器的很多资源，其次，应该对上传操作的时间加以更改，否则文件在上传的途中会被程序自动取消。下面根据以上情况制作一个规范化的文件上传模块，运行结果如图 13.16 所示。

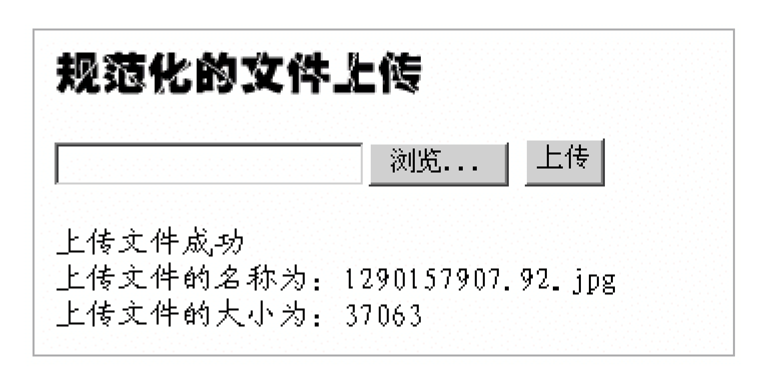


图 13.16　规范化的文件上传

操作步骤如下：

（1）创建脚本文件 index.php。编写文件上传域和一个提交按钮。

（2）当“上传”按钮被单击时，首先获取上传文件的相关信息并创建指定文件夹。其次，定义上传文件的名称。这里分别运用了时间戳函数和字符串截取函数，重新定义上传文件的名称。最后，利用 move_uploaded_file()函数将上传文件移动到指定文件夹。代码如下：

```
<?php
    if($_POST[sub]){                                    //“上传”按钮被单击
        $file=$_FILES['filename'];                      //获取上传文件的相关信息
        if(!is_dir("uploads")){                         //判断指定文件夹是否存在
            mkdir("uploads");                           //创建文件夹
        }
        if($file['size']>1024*1000){                    //限制上传文件的大小
            echo "上传文件超出指定大小";
        }else{
            $time=time();                               //获取当前时间戳
            $mic=microtime();                           //获取当前时间戳微秒数
```

```
                $name=$time+$mic;
                $string=substr($file['name'],-4,4);                   //截取字符串
                $name=$name.$string;                                  //拼接字符串
                $path="uploads/".$name;                               //拼接路径
                if(is_uploaded_file($file['tmp_name'])){              //判断上传的文件是否是通过 post 上传
                    if(move_uploaded_file($file['tmp_name'],$path)){  //移动到指定文件夹
                        echo "上传文件成功<br>";
                        echo "上传文件的名称为：".$name."<br>";
                        echo "上传文件的大小为：".$file['size'];
                    }
                }else{
                    echo "非法的文件上传";
                }
            }
        }
?>
```

DIY：在上面的应用中再添加一个输出文件的临时名称的操作，使读者了解文件名称更改的全过程。（20 分）（实例位置：光盘\mr\13\qjyy\01_diy）

13.5.2 情景应用 2——规范化的多文件上传

视频讲解：光盘\mr\13\lx\规范化的多文件上传.exe

实例位置：光盘\mr\13\qjyy\02

多文件上传，其实现原理与单文件上传是完全相同的，区别是多文件上传需要利用循环语句进行控制。笔者所制作的多文件上传允许单一文件上传和多文件上传两种。本实例的运行结果如图 13.17 所示。



图 13.17　多文件上传

操作步骤如下：

（1）创建脚本文件 index.php，创建文件上传域。

（2）当“上传”按钮被单击时，利用 for 循环语句循环获取上传文件的相关信息，并利用与 13.5.1 小节同样的方法实现规范化上传。代码如下：

```
<?php
        for($a=0;$a<4;$a++){                                  //循环语句
    ?>
            <input type="file" name="filename_<?php echo $a;?>">  //循环输出文件域
    <?php
```

```
            }
        ?>
        <br><input class="one"type="submit"name="sub" value="上传"/>  //提交按钮
        </form>
<?php
    if(isset($_POST[sub])){
        for($a=0;$a<4;$a++){                                   //循环语句
            $name=$_FILES['filename_'.$a];                     //将变量的名称保存在变量中
            $rand=rand(1,100000);                              //获取随机数
            $name_type=substr($name['name'],-4,4);             //获取文件名称的后 4 位即文件的格式
            $time=microtime();                                 //获取随机数
            $path='upload/'.(($rand+$time).$name_type);        //定义上传文件的路径
            $dir='upload/';                                    //定义目录
            if(move_uploaded_file($name['tmp_name'],$path)){
                echo "成功上传文件".($rand+$time).$name_type."<br>";   //输出提示
            }           //移动到服务器指定位置
        }
    }
?>
```

DIY：在上述实例的基础上，通过隐藏域 MAX_FILE_SIZE 的值对上传文件的大小进行控制。提示：必须将隐藏域放置在上传文件的文件域之前，否则是不会起到作用的。（20 分）（实例位置：光盘\mr\13\qjyy\02_diy）

13.5.3　情景应用 3——header()函数执行文件下载

视频讲解：光盘\mr\13\lx\header()函数执行文件下载.exe

实例位置：光盘\mr\13\qjyy\03

文件下载相信用户们并不陌生，一般情况下 PHP 中的文件下载分为两种方法：一是使用<a>标签实现压缩包下载。二是使用 header()函数实现下载。使用<a>标签下载具有局限性，因为它只能针对压缩包实现下载。本实例中应用 header()函数实现文件的下载，如果下载的文件不存在，则会给出提示信息。运行结果如图 13.18 所示。

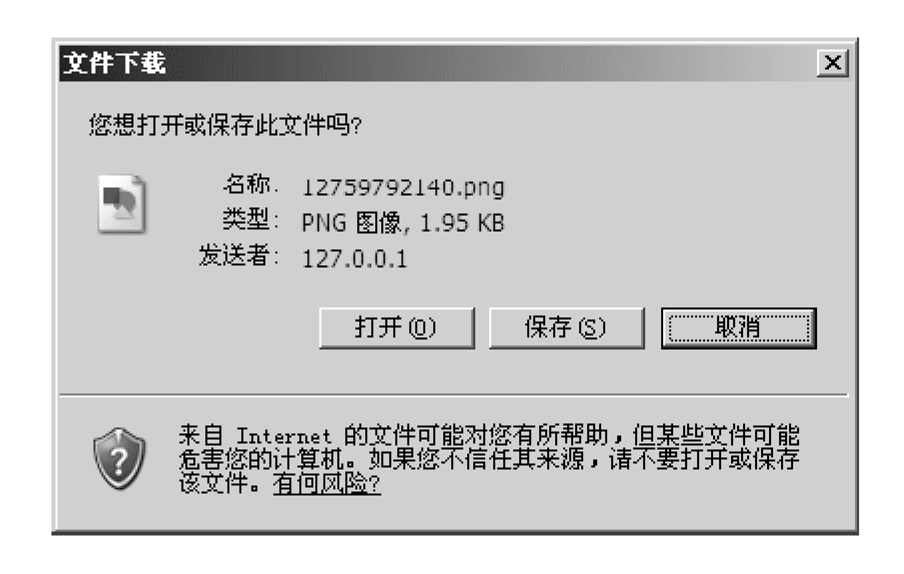




图 13.18　header()函数文件下载

（1）在 index.php 文件中，为输出的图片创建超链接，链接到 download.php 文件，通过 download.php 完成文件下载的操作。在 index.php 文件中创建超链接的代码如下：

```
<td width="245" height="100" align="center" valign="middle" bgcolor="#F0F0F0"><a href="download-<?php echo $myrow[file_test];?>.html" title="点击即可下载！"><?php echo "<img src=\"$myrow[file_test]\" width=\"250\" height=\"100\" border=\"0\">"?></a></td>
```

（2）创建 download.php 文件，应用 header()函数，实现文件的下载。代码如下：

```
<?php
	header("Content-type: text/html; charset=UTF-8");			//设置文件编码格式
	$path = $_GET['path'];							//获取文件路径
	if(!empty($path) and !is_null($path)){					//判断变量是否为空，是否为 null
		$filename=basename($path);					//获取文件名
		$file=@fopen($path,"r");
		if($file){
			header("Content-type:application/octet-stream");	//输出 MIME 类型
			header("Accept-ranges:bytes");				//接受的范围单位
			header("Accept-length:".filesize($path));		//文件长度
			header("Content-Disposition:attachment;filename=".$filename);
										//默认时文件保存对话框中的文件名称
			echo fread($file,filesize($path));			//读取文件
			fclose($file);						//关闭文件
			exit;							//退出
		}else{
			echo "<script>alert('您下载的文件不存在！'); history.back();</script>";
		}
	}
?>
```

DIY：使用<a>标签实现压缩包的下载。（20 分）（实例位置：光盘\mr\13\qjyy\03_diy）

13.5.4 情景应用 4——文本文件的分页输出

视频讲解：光盘\mr\13\lx\文本文件的分页输出.exe

实例位置：光盘\mr\13\qjyy\04

在开发程序过程中，有时会应用到大量的文字信息，如果文字信息的数量过多最好是使用分页来显示。在每一页中显示指定数量的文字信息，如果超过指定数量就在下一页中显示。本实例就是通过分页来显示文字信息的，运行结果如图 13.19 所示。

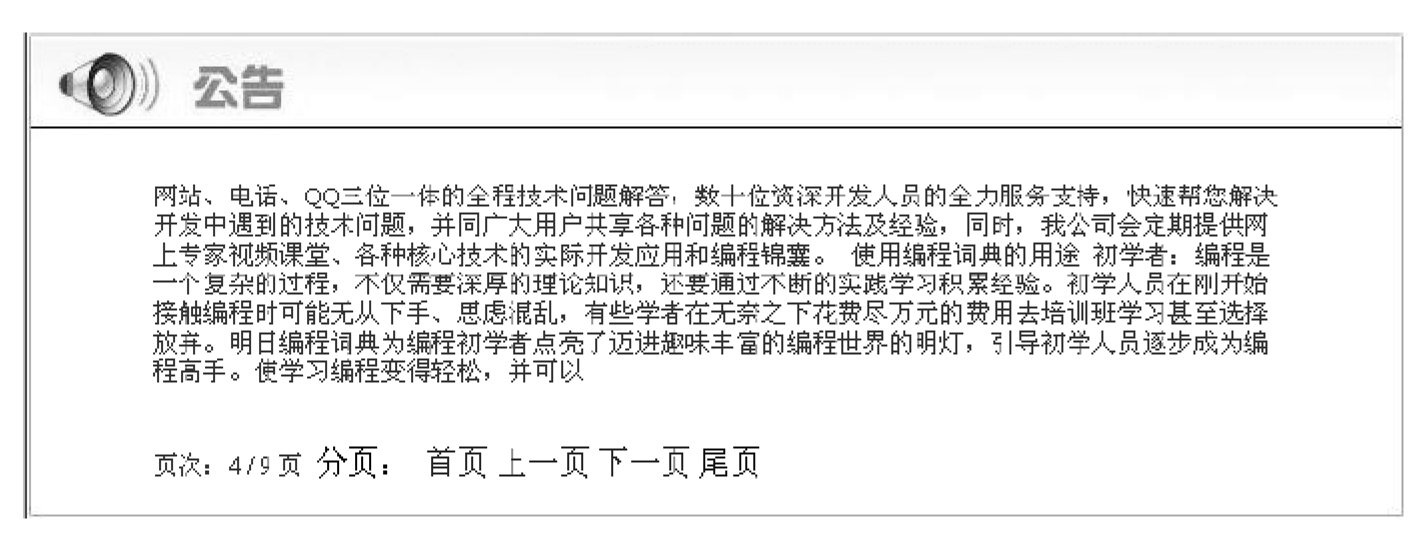


图 13.19　超长文件分页显示

完成超长文本的分页输出需要 3 方面的技术：第一方面，自定义函数，通过自定义函数读取文本文件，可以避免中文字符串出现乱码；第二方面，字符串函数，需要通过 strlen()函数计算字符串的长度，通过 substr()函数对字符串进行截取；第三方面，文件系统函数，通过 file_get_contents()函数读取文本文件中的数据。

自定义函数 msubstr()的语法如下：

```
function msubstr($str,$start,$len){		//$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
	$tmpstr="";
```

```
        $strlen=$start+$len;                    //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
        for($i=0;$i<$strlen;$i++){              //通过 for 循环语句循环读取字符串
                if(ord(substr($str,$i,1))>0xa0){  //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
                        $tmpstr.=substr($str,$i,2); //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
                        $i++;                   //变量自加 1
                }else{                          //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
                        $tmpstr.=substr($str,$i,1);
                }
        }
        return $tmpstr;                         //输出字符串
}
```

具体步骤如下：

（1）创建 function.php 文件，编写自定义函数 msubstr()，完成对文本文件的截取操作。

（2）创建 index.php 文件，首先通过文件系统函数 file_get_contents()读取整个文件的内容，然后调用自定义函数和字符串函数完成对文件的截取操作，实现截取后内容的分页输出。关键代码如下：

```
<?php
if (isset($_GET['page'])) {                     //定义初识变量的值
        $page=$_GET['page'];
}else{
        $page=1;                                //变量值为 1
}
include("function.php");                        //调用自定义函数文件
?>
   <?php
//读取超长文本中的数据，实现超长文本中数据的分页显示
if($page){
        $counter=file_get_contents("file/file.txt");  //读取文本文件
        $length=strlen(unhtml($counter));       //获取文本文件的长度，并通过自定义函数去除特殊字符和空格
        $page_count=ceil($length/850);          //对超长文本进行分页，每页显示 850 个字符
        $c=msubstr($counter,0,($page-1)*850);   //计算上一页的字节
        $c1=msubstr($counter,0,$page*850);      //计算下一页的字节
                echo substr($c1,strlen($c),strlen($c1)-strlen($c));    //获取当前的字节
}
?></td>
            <td rowspan="2"> </td>
          </tr>
          <tr>
            <td><span class="STYLE3">页次：<?php echo $page;?> / <?php echo $page_count;?>  页  </span>
          分页：
                <?php
                    if($page!=1){
                            echo   "<a href=index.php?page=1>首页</a> ";
                             echo "<a href=index.php?page=".($page-1).">上一页</a> ";
                        }
                        if($page<$page_count){
                                echo "<a href=index.php?page=".($page+1).">下一页</a> ";
                                echo   "<a href=index.php?page=".$page_count.">尾页</a>";
                        }
            ?>
```

DIY：改变当页输出字符串的数量实现超长文本的分页显示。(20 分)(实例位置：光盘\mr\13\qjyy\04_diy)

13.5.5 情景应用 5——将数据以二进制格式上传到数据库

视频讲解：光盘\mr\13\lx\将数据以二进制格式上传到数据库.exe

实例位置：光盘\mr\13\qjyy\05

在网站开发过程中，有一个问题是必须考虑到的，网站是否支持文件上传的功能？如果支持把文件存储于什么地方？是服务器中，还是存储在数据库中。针对论坛中读者的一些意见和帖子还是保存在数据库中比较方便，同时还可以对它们进行及时的维护。下面就介绍一个以二进制形式上传文件到数据库的例子，运行本实例，单击“浏览”按钮，在本机上查找要上传的文件，然后打开该文件，最后单击“提交”按钮将文件以二进制的形式保存到数据库，本实例的运行结果如图 13.20 所示。



图 13.20　以二进制形式上传文件到数据库

本实例通过表单中的文件域选择框选择要上传的文件，然后将文件以二进制的形式保存到数据库中。将表单提交的数据存储到数据库中，主要使用 Insert 语句，语法格式如下：

```
Insert into table_name (column_name,column_name2, … ) values (value1, value2, … )
```

参数说明：

- ☑ table_name：数据表的名称。
- ☑ column：数据表中字段的名称。
- ☑ valuel：表单提交的变量。

具体步骤如下：

（1）既然是将文件存储到数据库中，首先应该与数据库建立连接，连接数据库的代码如下：

```
<?php
$id=mysql_connect('localhost','root','111');                    //连接数据库
mysql_select_db("db_database13",$id);
mysql_query("set names utf8");
?>
```

（2）然后创建表单来提交数据，将数据提交到 index_ok.php 文件中，并使用 JavaScript 脚本对提交的数据进行验证，部分代码如下：

```
<form name="form1" method="post" action="index_ok.php">
  <tr>
    <td height="28"> </td>
    <td align="left" valign="middle"><input name="file" type="file" size="22"></td>
    <td> </td>
  </tr>
  <tr>
    <td height="30"> </td>
    <td align="center" valign="top"><input type="submit" name="Submit" value="提交" onClick="file1(form1);"></td>
    <td> </td>
  </tr>
</form>
```

（3）应用 Insert 语句将用户提交的数据以二进制的形式存储到数据库中，代码如下：

```
<?php
header('Content-Type:text/html;charset=utf-8');
```

```
session_start();
include("conn/conn.php");
$data=date("Y-m-d");
$file1="file_name1";
if(isset($_POST['Submit'])){
	$query=mysql_query("Insert into tb_file1 (file_name,file_text,data)values('$file1','".$_POST['file']."','$data')");
	if($query){
		echo "文件上传成功!!";
		echo "<meta http-equiv=\"refresh\" content=\"3 url=index.php\">";
	}else{
		echo "上传失败!!";
		echo "<meta http-equiv=\"refresh\" content=\"3 url=index.php\">";
	}
}
?>
```

DIY：将上面的实例更改为将图片地址保存到数据库中并实现预览。（20 分）（实例位置：光盘\mr\13\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

13.6　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．判断某目录是否存在需要使用函数（　　）。

A．is_float　　B．is_int　　C．is_dir

2．读取目录结构的函数为（　　）。

A．opendir　　B．closedir　　C．scandir　　D．file

3．下面（　　）变量可以获取文件的临时名称。

A．$_FILES　　B．$_POST　　C．$_GET

4．开启文件上传需要修改（　　）文件。

A．phpinfo　　B．php.ini　　C．php.txt

5．创建指定文件目录需要使用的函数为（　　）。

A．open　　B．is_dir　　C．mkdir

二、填空题（每题 10 分，5 道题）

1．实现将指定文件(name="file")上传并将文件相关信息保存在数组$array 中。请将代码补充完整。

```
		If(isset($_POST[sub])){
（                                        ）
		}
```

2．将上传文件移动到指定文件夹需要使用函数（　　　　）。

3．删除目录的函数是（　　　　）。

4．判断指定的文件是否是通过 HTTP POST 上传的需要使用函数（　　）。

5．本部分代码实现的功能是（　　）。

```
header('Content-Type:image/jpg');
header('Content-Disposition:attachment;filename="test.jpg"');
header('Content-Length:'.filesize('test.jpg'));
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

13.7 行 动 指 南

开始日期：______年____月____日　　　　　　　　结束日期：______年____月____日

<table>
<tr><th>序号</th><th>内　容</th><th colspan="2">行 动 指 南</th></tr>
<tr><td rowspan="9">1</td><td rowspan="3">照猫画虎栏目
分数（　　）</td><td>分数>75 分</td><td>优秀，基本功掌握得不错，加油！</td></tr>
<tr><td>75 分>分数>50 分</td><td>及格，知识掌握得不牢，重新做一遍照猫画虎。</td></tr>
<tr><td>分数<50 分</td><td>请使用光盘中提供的“实战能力测试系统”进行提高训练。</td></tr>
<tr><td rowspan="3">情景应用栏目
分数（　　）</td><td>分数>75 分</td><td>优秀，综合应用能力很强。</td></tr>
<tr><td>75 分>分数>50 分</td><td>及格，综合应用能力需提高，再练一遍情景应用。</td></tr>
<tr><td>分数<50 分</td><td>请使用光盘中提供的“实战能力测试系统”进行提高训练。</td></tr>
<tr><td rowspan="2">自我测试栏目
分数（　　）</td><td>分数>75 分</td><td>优秀，有成为编程高手的潜质。</td></tr>
<tr><td>分数<75 分</td><td>请使用光盘中提供的“实战能力测试系统”进行提高训练。</td></tr>
<tr><td>综合评价</td><td colspan="2">反复训练后，以上各项分数都在 75 分以上，方可以进入下一堂课学习。</td></tr>
<tr><td rowspan="4">2</td><td rowspan="4">编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。</td><td colspan="2">（1）制作一个简单的文档管理系统。</td></tr>
<tr><td colspan="2">（2）制作规范化上传文件模块。</td></tr>
<tr><td colspan="2">（3）将上传文件移动到服务器下指定文件夹。</td></tr>
<tr><td colspan="2">（4）制作一个为用户提供文件下载的操作。</td></tr>
<tr><td rowspan="3">3</td><td rowspan="3">编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。</td><td colspan="2"></td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2"></td></tr>
<tr><td rowspan="4">4</td><td rowspan="4">创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。</td><td colspan="2"></td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2"></td></tr>
</table>

13.8　成功可以复制——知识改变命运、科技改变生活李彦宏

成就：最先创建了 ESP 技术，并拥有“超链分析”技术的美国专利，是奠定整个现代搜索引擎发展趋势和方向的基础发明之一。

创始公司：中国最大的搜索引擎公司——百度网络技术有限公司

财富巅峰：胡润 2010 百富榜，李彦宏以身价 360 亿成为国内 IT 界首富。

最高学历：硕士

出生日期：1968 年 11 月

超级语录：“活的搜索，改变生活。”

成功历程

19 岁的李彦宏离开山西阳泉到梦想中的北大就读信息管理专业，23 岁远渡重洋赴美国布法罗纽约州立大学主攻计算机，正是北大的信息管理专业让他深谙搜索内涵，正是美国的计算机学业让他掌握计算机工具。

在搜索引擎发展初期，李彦宏作为全球最早研究者之一，最先创建了 ESP 技术，并将它成功地应用于 Infoseek/的搜索引擎中。另外，图像搜索引擎是他的另一项极具应用价值的技术创新。



在李彦宏 31 岁时，创建中国最大的搜索引擎公司——百度网络技术有限公司。知识改变了命运！百度公司创始人、CEO 李彦宏十分相信这句话。

在美国的 8 年人生历程中，李彦宏先后担任了道·琼斯公司高级顾问、《华尔街日报》网络版实时金融信息系统设计者、国际知名互联网企业-Infoseek 资深工程师等职务。他为道·琼斯公司设计的实时金融系统，迄今仍被广泛地应用于华尔街各大公司的网站。

1999 年底，李彦宏携 120 万美金的风险投资回国与好友徐勇先生共同创建百度网络技术有限公司，并在短短 6 个月的时间内完成目前中国最大、最好的中文搜索引擎的开发工作。

2005 年 8 月，百度在美国纳斯达克成功上市，成为全球资本市场最受关注的上市公司之一。目前，百度也是全球跨国公司最多寻求合作的中国公司，随着百度日本公司的成立，百度加快了走向国际化的步伐。

深度评价

从李彦宏的求学、工作及创业经历中给我们这样的启示，知识可以改变命运，知识可以改变生活。作为程序设计人员，要多学习，刻苦钻研，掌握更多的知识和技能，然后把学到的知识和技能应用到实际工作和生活中，从而去改变我们的工作和生活。

第 3 部分

高级篇

第14堂课

面向对象的程序设计

（视频讲解：180分钟）

面向对象（OOP）的编程方式是PHP的突出特点之一，采用这种编程方式可以对大量零散代码进行有效组织，从而使PHP具备大型Web项目开发的能力。采用面向对象编程方式还可以提高网站的易维护性和易读性。总之，面向对象的编程方式是计算机发展史上具有划时代意义的标志。习惯了面向过程编程思想的程序员，在刚接触面向对象的编程方式时，可能会感觉困难，毕竟两者在开发程序时，无论是方式还是方法都是截然不同的。那么，如何去应对呢？答案是多学习、勤思考，在学习面向对象的编程方式过程中，还应该多读已成型的、优秀的程序，从而全面提高个人的编程能力。以全面掌握面向对象的编程思想为出发点，兼顾面向过程和面向对象两种编程思想为目的，本堂课将采用讲解与实例相结合的方式、由浅入深，最终使读者能够将面向对象的编程思想应用到实际项目开发中去。

学习摘要：

- 一切皆是对象
- 类的声明
- 类的实例化
- 面向对象的封装特性
- 面向对象的继承特性
- 抽象类和接口
- 面向对象的多态性
- 面向对象的关键字
- 面向对象的魔术方法

14.1 一切皆是对象

面向对象就是将要处理的问题抽象为对象，然后通过对象的属性和行为来解决对象的实际问题。面向对象的基本概念就是类和对象，接下来将分别进行讲解。

14.1.1 什么是类

正所谓：“物以类聚，人以群分”。世间万物都具有其自身的属性和方法，通过这些属性和方法可以将不同物质区分开来。例如，人具有性别、体重和肤色等属性，还可以进行吃饭、睡觉、学习等能动活动，这些活动可以说是人具有的功能。可以把人看作程序中的一个类，那么人的性别可以比作类中的属性，吃饭可以比作类中的方法。

也就是说，类是属性和方法的集合，是面向对象编程方式的核心和基础，通过类可以将零散的用于实现某项功能的代码进行有效管理。例如，创建一个数据库连接类，包括 6 个属性，即数据库类型、服务器、用户名、密码、数据库和错误处理；包括 3 个方法，即定义变量方法、连接数据库方法和关闭数据库方法。数据库连接类的设计效果如图 14.1 所示。

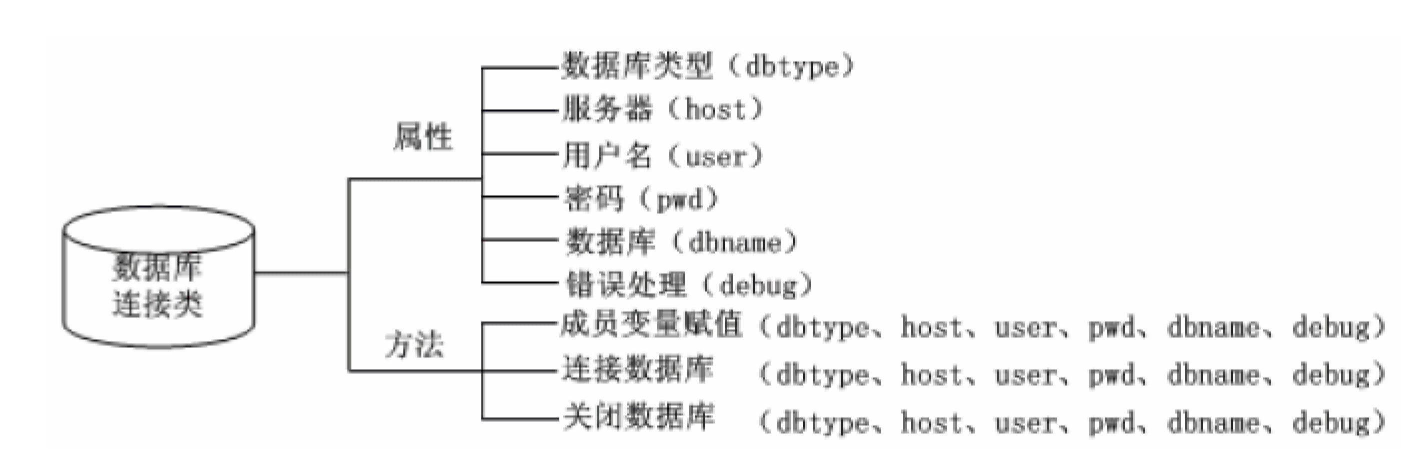


图 14.1　数据库连接类的设计效果

14.1.2 对象的由来

类只是具备某项功能的抽象模型，实际应用中还需要对类进行实例化，这样就引入了对象的概念。对象是类进行实例化后的产物，是一个实体。仍然以人为例，“黄种人是人”这句话没有错误，但反过来说“人是黄种人”这句话一定是错误的。因为除了有黄种人，还有黑人、白人等。那么“黄种人”就是“人”这个类的一个实例对象。可以这样理解对象和类的关系：对象实际上就是“有血有肉的、能摸得到看得到的”一个类。

这里实例化创建的数据库连接类，调用数据库连接类中的方法，完成与数据库的连接操作，如图 14.2 所示。

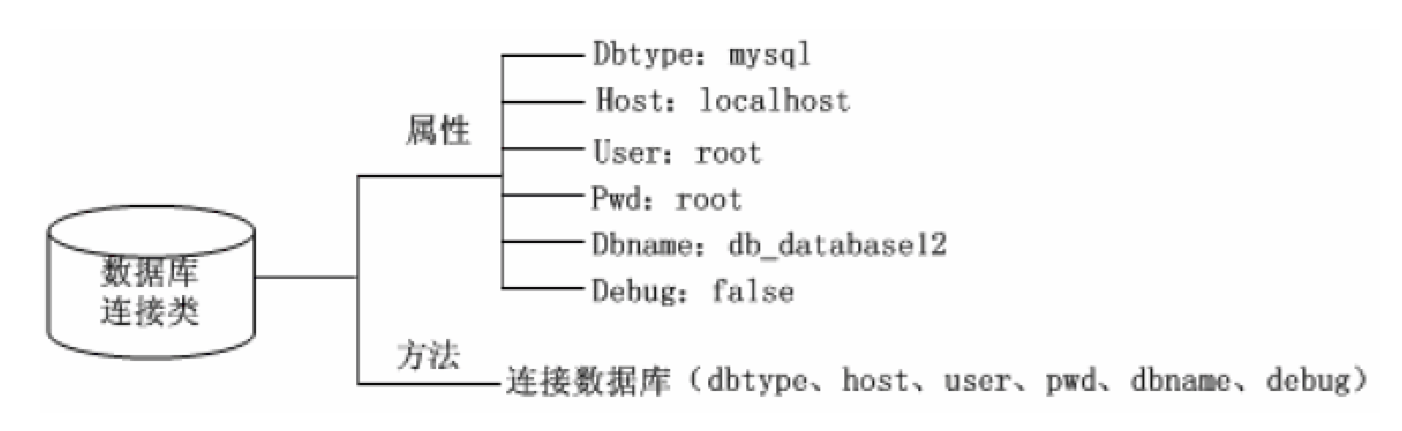


图 14.2　实例化对象

14.1.3　面向对象的特点

面向对象编程的 3 个重要特点是继承、封装和多态，它们迎合了编程中注重代码重用性、灵活性和可扩展性的需要，奠定了面向对象在编程中的地位。

（1）封装性：就是将一个类的使用和实现分开，只保留有限的接口（方法）与外部联系。对于使用该类的开发人员，只要知道这个类该如何使用即可，而不用去关心这个类是如何实现的。这样做可以让开发人员更好地把精力集中起来专注别的事情，同时也避免了程序之间的相互依赖而带来的不便。

例如，使用电脑时，并不需要将电脑拆开了解每个部件的具体用处，用户只需按下主机箱上的 Power 按钮就可以启动电脑。但对于电脑内部的构造，用户可以不必了解，这就是封装的具体表现。

（2）继承性：是派生类（子类）自动继承一个或多个基类（父类）中的属性与方法，并可以重写或添加新的属性或方法。继承这个特性简化了对象和类的创建，增加了代码的可重用性。

假如已经定义了 A 类，接下来准备定义 B 类，而 B 类中有很多属性和方法与 A 类相同，那么就可以使 B 类继承于 A 类，这样就无须再在 B 类中定义 A 类已有的属性和方法，从而可以在很大程度上提高程序的开发效率。

例如，定义一个水果类，水果类具有颜色属性，然后定义一个苹果类，在定义苹果类时完全可以不定义苹果类的颜色属性，通过如图 14.3 所示继承关系完全可以使苹果类具有颜色属性。

（3）多态性：指同一个类的不同对象，使用同一个方法可以获得不同的结果。多态性增强了软件的灵活性和重用性。

例如，定义一个火车类和一个汽车类，火车和汽车都可以移动，说明两者在这方面可以进行相同的操作，然而，火车和汽车移动的行为是截然不同的，因为火车必须在铁轨上行驶，而汽车在公路上行驶，这就是类多态性的形象比喻，如图 14.4 所示。

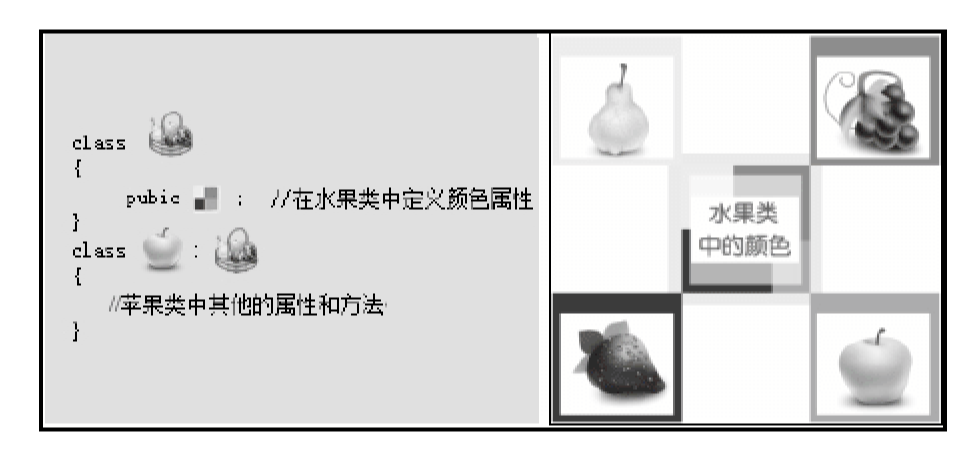


图 14.3　继承特性效果示意图

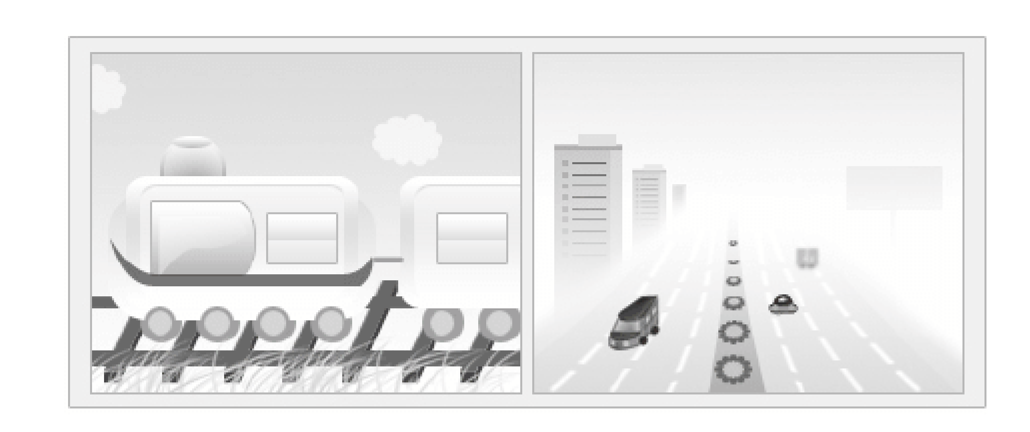


图 14.4　多态在生活中的体现

14.2　类 的 声 明

在创建类名称时必须将类进行声明。

14.2.1　类的定义

和很多面向对象的语言一样，PHP 也是通过 class 关键字加类名来定义类的。类的格式如下：

```
<?php
    权限修饰符　class　类名{
        类体;
```

```
    }
?>
```

- ☑ 权限修饰符是可选项，可以使用 public、protected、private 或者省略这 3 者。
- ☑ class 是创建类的关键字。
- ☑ 类名是所要创建类的名称，必须写在 class 关键字之后，在类的名称后面必须跟上一对大括号。
- ☑ 类体是类的成员，类体必须放在类名后面的两个大括号“{”和“}”之间。

说明： 在创建类时，在 class 关键字前除可以加权限修饰符外，还可以加其他关键字如 static、abstract 等，有关创建类使用的权限修饰符和其他关键字将在后面的内容中进行讲解。至于类名的定义，与变量名和函数名的命名规则类似，如果由多个单词组成，习惯上每个单词的首字母要大写，并且类名应该有一定的意义。

例如，创建一个 ConnDB 类。代码如下：

```
<?php
class ConnDB{                        //定义数据库连接类
    //…
}
?>
```

注意： 虽然 ConnDB 类仅有一个类的骨架，什么功能都没有实现，但这并不影响它的存在。一个类即一对大括号之间的全部内容都要在一段代码段中，不允许将类中的内容分割成块，例如：

```
<?php
class ConnDB{                        //定义数据库连接类
    //…
?>
<?php
    //…
}
?>
```

这种格式是不允许的。

14.2.2 成员属性

在类中直接声明的变量称为成员属性（也可以称为成员变量），可以在类中声明多个变量，即对象中有多个成员属性，每个变量都存储对象不同的属性信息。

成员属性的类型可以是 PHP 中的标量类型和复合类型，但是如果使用资源和空类型是没有意义的。

成员属性的声明必须由关键字来修饰，如 public、protected、private 等，这是一些具有特定意义的关键字。如果不需要有特定的意义，那么可以使用 var 关键字来修饰。还有就是在声明成员属性时没有必要赋初始值。

下面再次创建 ConnDB 类并在类中声明一些成员属性，其代码如下：

```
class ConnDB{                        //定义类
    var  $dbtype;                    //声明成员属性
    var  $host;                      //声明成员属性
    var  $user;                      //声明成员属性
```

```
        var   $pwd;                    //声明成员属性
        var   $dbname;                 //声明成员属性
        var   $debug;                  //声明成员属性
        var   $conn;                   //声明成员属性
}
```

14.2.3　成员方法

在类中声明的函数称为成员方法。一个类中可以声明多个函数，即对象中可以有多个成员方法。成员方法的声明和函数的声明相同，唯一特殊之处是成员方法可以由关键字来对它进行修饰，控制成员方法的权限。声明成员方法的代码如下：

```
class ConnDB{                          //定义类
        function ConnDB(){             //声明构造方法
                //方法体
        }
        function GetConnId(){          //声明数据库连接方法
                //方法体
        }
        function CloseConnId(){        //声明数据库关闭方法
                $this->conn->Disconnect();   //方法体，执行关闭的操作
        }
}
```

在类中成员属性和成员方法的声明都是可选的，可以同时存在，也可以单独存在。具体应该根据实际的需求而定。

14.3　类的实例化

14.3.1　创建对象

面向对象程序的最终操作者是对象，而对象是类实例化的产物。所以学习面向对象只停留在类的声明上是不够的，必须学会将类实例化成对象。类的实例化格式如下：

```
$变量名=new 类名称([参数]);           //类的实例化
```

- ☑ $变量名：类实例化返回的对象名称，用于引用类中的方法。
- ☑ new：关键字，表明要创建一个新的对象。
- ☑ 类名称：表示新对象的类型。
- ☑ 参数：指定类的构造方法用于初始化对象的值。如果类中没有定义构造函数，PHP 会自动创建一个不带参数的默认构造函数。

例如，这里对上面创建的 ConnDB 类进行实例化。代码如下：

```
class ConnDB{                          //定义类
        function ConnDB(){             //声明构造方法
                //方法体
        }
        function GetConnId(){          //声明数据库连接方法
                //方法体
```

```
    }
    function CloseConnId(){                    //声明数据库关闭方法
        $this->conn->Disconnect();                     //方法体，执行关闭的操作
    }
}
$connobj1=new ConnDB();                              //类的实例化
$connobj2=new ConnDB();                              //类的实例化
$connobj3=new ConnDB();                              //类的实例化
```

一个类可以实例化多个对象，每个对象都是独立的。如果上面的 ConnDB 类实例化了 3 个对象，就相当于在内存中开辟了 3 个空间存放对象。同一个类声明的多个对象之间没有任何联系，只能说明它们是同一个类型。就像是 3 个人，都有自己的姓名、身高、体重，都可以进行吃饭、睡觉、学习等活动。

14.3.2 访问类中成员

在类中包括成员属性和成员方法，访问类中的成员包括成员属性和方法的访问。访问方法与访问数组中的元素类似，需要通过对象的引用来访问类中的每个成员。其中还要应用到一个特殊的运算符号“->”。访问类中成员的语法格式如下：

```
$变量名=new 类名称([参数]);                          //类的实例化
$变量名->成员属性=值;                                //为成员属性赋值
$变量名->成员属性;                                   //直接获取成员属性值
$变量名->成员方法;                                   //访问对象中指定的方法
```

这是访问类中成员的基本格式，下面看它们在具体的实例中是如何运用的。

例 14.01 创建 ConnDB 类，对类进行实例化，并访问类中的成员属性和成员方法。代码如下：**（实例位置：光盘\mr\14\sl\14.01）**

```
<?php
    class mysql{                                     //定义数据库连接类
        var $localhost;                              //定义成员变量
        var $name;
        var $pwd;
        var $db;
        var $conn;
        public function mysql($localhost,$name,$pwd,$db){   //定义构造方法
            $this->localhost=$localhost;             //为成员变量赋值
            $this->name=$name;
            $this->pwd=$pwd;
            $this->db=$db;
            $this->connect();
        }
        public function connect(){                   //定义数据库连接方法
            $this->conn=$conn;
            $this->conn=mysql_connect($this->localhost,$this->name,$this->pwd)or die("CONNECT MYSQL
FALSE");                                             //执行连接操作
            mysql_select_db($this->db,$this->conn)or die("CONNECT DB FALSE");  //选择数据库
            mysql_query("SET NAMES utf8");           //设置数据库编码格式
        }
        public function GetId(){                     //定义方法，返回数据库连接信息
            echo "MYSQL 服务器的用户名：".$this->name."<br>";
```

```
                echo "MYSQL 服务器的密码：".$this->pwd;
            }
        }
        $msl=new mysql("127.0.0.1","root","111","db_database14");     //实例化数据库连接类
        $msl->GetId();                                                 //调用类中方法
?>
```

本实例的运行结果如图 14.5 所示。

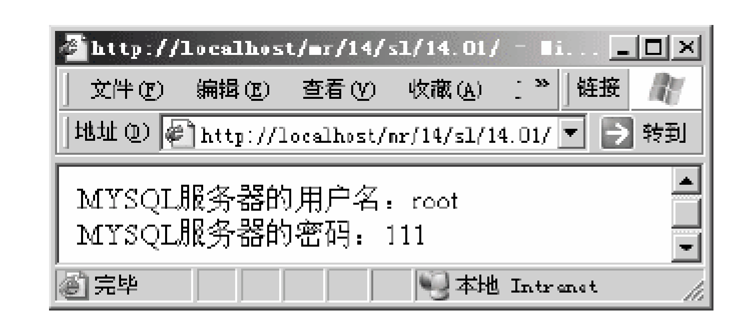


图 14.5　访问类中成员的结果

14.3.3　特殊的访问方法——“$this”和“::”

1．$this

在例 14.01 中，使用了一个特殊的对象引用方法“$this”。那么它到底表示什么意义呢？在这里将进行详细的讲解。

$this 存在于类的每个成员方法中，它是一个特殊的对象引用方法。成员方法属于哪个对象，$this 引用就代表哪个对象，其作用就是专门完成对象内部成员之间的访问。

正如在例 14.01 中定义的那样，将传递的参数值直接赋给成员变量，而在 GetConnId()方法中，直接通过$this->user 和$this->pwd 获取数据库的用户名和密码。

2．操作符“::”

相比$this 引用只能在类的内部使用，操作符“::”才是真正的强大。操作符“::”可以在没有声明任何实例的情况下访问类中的成员。例如，在子类的重载方法中调用父类中被覆盖的方法。操作符“::”的语法格式如下：

```
关键字::变量名/常量名/方法名
```

这里的关键字分为以下 3 种情况。

☑　parent 关键字：可以调用父类中的成员变量、成员方法和常量。

☑　self 关键字：可以调用当前类中的静态成员和常量。

☑　类名：可以调用本类中的变量、常量和方法。

例 14.02　本实例依次使用类名、parent 关键字和 self 关键字来调用变量和方法。读者可以观察输出的结果。代码如下。**（实例位置：光盘\mr\14\sl\14.02）**

```
<?php
/*
        当实例化对象后不需要使用对象句柄调用对应的方法时可以只给类实例化不返回对象句柄。
*/
    class Car{
        const NAME="别克系列";
        public function __construct(){              //定义构造方法
            echo "父类：".Car::NAME;                //类名引用
        }
```

```
	}
	class SmallCar extends Car{                        //继承
		const NAME="别克君威";
		public function __construct(){                 //定义构造方法
			echo parent::__construct()."\t";          //应用父类构造方法
			echo "子类：".self::NAME;
		}
	}
	new SmallCar();                                    //实例化对象
?>
```

运行结果如图 14.6 所示。

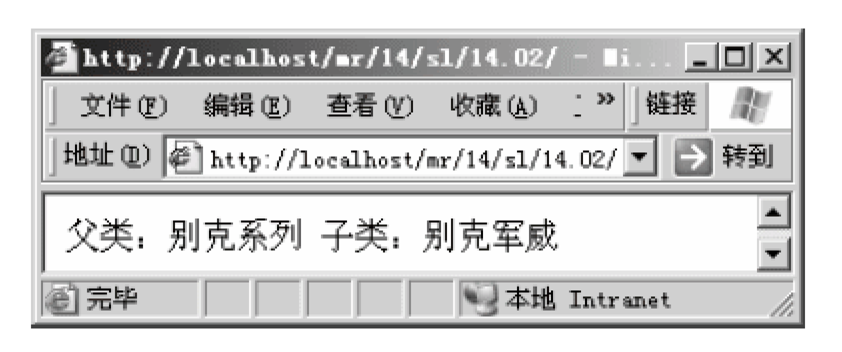


图 14.6　关键字的使用

14.3.4　构造方法和析构方法

1．构造方法

构造方法是对象创建完成后第一个被对象自动调用的方法。它存在于每个声明的类中，是一个特殊的成员方法，如果在类中没有直接声明构造方法，那么类中会默认生成一个没有任何参数且内容为空的构造方法。

构造方法多数是执行一些初始化的任务。例如，例 14.01 中通过构造方法为成员变量赋初始值。

在 PHP 中，构造方法的声明有两种情况：第一种在 PHP 5 以前的版本中，构造方法的名称必须与类名相同；第二种在 PHP 5 的版本中，构造方法的方法名称必须是以两个下划线开头的，即“__construct()”。虽然在 PHP 5 中构造方法的声明方法发生了变化，但是以前的方法还是可用的。

PHP 5 中的这个变化是考虑到构造函数可以独立于类名，当类名发生变化时不需要修改相应的构造函数的名称。通过__construct()声明构造方法的语法格式如下：

```
function __construct([mixed args [,…]]){
		//方法体
}
```

在 PHP 中，一个类只能声明一个构造方法。在构造方法中可以使用默认参数，实现其他面向对象的编程语言中构造方法重载的功能。如果在构造方法中没有传入参数，那么将使用默认参数为成员变量进行初始化。

例 14.03　在例 14.01 中，通过使用与类名相同的方法声明构造方法，那么这里将通过__construct()声明一个与类名不同的构造方法。代码如下。**（实例位置：光盘\mr\14\sl\14.03）**

```
<?php
/*
		构造函数：当类被实例化后构造函数自动执行。所以如果用户希望在实例化的同时调用某个方法可以通过 this 关键字调用。
*/
	class mysql{                         //定义类名称
```

```
            var $localhost;                    //定义变量
            var $name;
            var $pwd;
            var $db;
            var $conn;
            public function __construct($localhost,$name,$pwd,$db){        //构造函数
                $this->localhost=$localhost;
                $this->name=$name;
                $this->pwd=$pwd;
                $this->db=$db;
                $this->connect();
            }
            //省略了部分代码
        }
        $msl=new mysql("127.0.0.1","root","111","db_database14");     //实例化对象
        $msl->GetId();                                                 //对象句柄调用指定的方法
?>
```

其运行结果与例 14.01 相同。

2．析构方法

析构方法的作用和构造方法正好相反，是对象被销毁之前最后一个被对象自动调用的方法。它是 PHP 5 中新添加的内容，实现在销毁一个对象之前执行一些特定的操作，如关闭文件、释放内存等。

析构方法的声明格式与构造方法类似，都是以两个下划线开头的，即“__destruct”，析构函数没有任何参数。其语法格式如下：

```
function __destruct(){
            //方法体，通常是完成一些在对象销毁前的清理任务
}
```

在 PHP 中，有一种“垃圾回收”机制，可以自动清除不再使用的对象，释放内存。而析构方法就是在这个垃圾回收程序执行之前被调用的方法，在 PHP 中它属于类中的可选内容。

14.4　面向对象的封装特性

面向对象编程的特点之一是封装性，将类中的成员属性和方法结合成一个独立的相同单位，并尽可能隐藏对象的内容细节。其目的就是确保类以外的部分不能随意存取类的内部数据（成员属性和成员方法）。从而有效避免外部错误对类内数据的影响。

类的封装是通过关键字 public、private、protected、static 和 final 来实现的。下面对其中的 public、private 和 protected 关键字进行详细讲解。

14.4.1　public（公共成员）

顾名思义，就是可以公开的、没有必要隐藏的数据信息。可以在程序的任何地点（类内、类外）被其他的类和对象调用。子类可以继承和使用父类中所有的公共成员。

在本堂课的前半部分，所有的变量都被声明为 public，而所有的方法在默认的状态下也是 public。所以对变量和方法的调用显得十分混乱。为了解决这个问题，就需要使用第二个关键字“private”。

14.4.2　private（私有成员）

被 private 关键字修饰的变量和方法，只能在所属类的内部被调用和修改，不可以在类外被访问，即使是子类中也不可以。

例 14.04　通过调用成员方法对私有变量$name 进行修改与访问；如果直接调用私有变量，将会发生错误。代码如下。**（实例位置：光盘\mr\14\sl\14.04）**

```
<?php
	class Car{
		private $carName="奥迪系列";				//定义私有变量并赋值
		public function setName($carName){			//利用 set 方法设置变量值
			$this->carName=$carName;
		}
		public function getName(){				//利用 get 方法返回变量值
			return $this->carName;
		}
	}
	class SmallCar extends Car{				//继承
	}
	$car=new SmallCar();					//实例化子类对象
	$car->setName("Q7");					//为子类变量赋值
	echo "正确操作私有变量<br>";
	echo $car->getName();					//输出子类变量的值
	echo "<br>错误操作私有变量";
	echo Car::$carName;
?>
```

运行结果如图 14.7 所示。

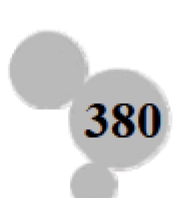


图 14.7　private 关键字

技巧：对于成员方法，如果没有写关键字，那么默认就是 public。从本节开始，以后所有的方法及变量都会带上关键字，这是作为一名程序员的一种良好的编程习惯。

14.4.3　protected（保护成员）

private 关键字可以将数据完全隐藏起来，除了在本类外，其他地方都不可以调用，子类也不可以。对于有些变量希望子类能够调用，但对另外的类来说，还要做到封装。这时，就可以使用 protected。被 protected 修饰的类成员，可以在本类和子类中被调用，其他地方则不可以被调用。



例 14.05　首先声明一个 protected 变量，然后使用子类中的方法调用，最后在类外直接调用一次。代码如下。（**实例位置：光盘\mr\14\sl\14.05**）

```
<?php
	class Car{                                         //定义轿车类
		protected $carName="奥迪系列";                 //定义保护变量
	}
	class SmallCar extends Car{                        //小轿车类定义轿车类
		public function say(){                         //定义 say 方法
			echo "调用父类中的属性：".$carName=$this->carName;  //输出父类变量
		}
	}
	$car=new SmallCar();                               //实例化对象
	$car->say();                                       //调用 say 方法
	$car->$carName='奥迪 Q7';
?>
```

运行结果如图 14.8 所示。



图 14.8　protected 关键字

技巧：虽然 PHP 中没有对修饰变量的关键字做强制性的规定和要求，但从面向对象的特征和设计方面考虑，一般使用 private 或 protected 关键字来修饰变量，以防止变量在类外被直接修改和调用。

14.5　面向对象的继承特性

面向对象编程的特点之二是继承性，使一个类继承并拥有另一个已存在类的成员属性和成员方法，其中被继承的类称为父类，继承的类称为子类。通过继承能够提高代码的重用性和可维护性。

14.5.1　类的继承——extends 关键字

类的继承是类与类之间的一种关系的体现。子类不仅有自己的属性和方法，而且还拥有父类的所有属性和方法，正所谓子承父业。

在 PHP 中，类的继承通过关键字 extends 实现，其语法格式如下：

```
class 子类名称 extends 父类名称{
	//子类成员变量列表
	function 成员方法(){                   //子类成员方法
		//方法体
	}
	//省略其他方法
}
```

读者应该记得在 14.1.3 小节中介绍面向对象的特点是，通过一个水果父类和一个苹果子类来形象地比喻面向对象继承性的特点。下面就创建这个子类和父类，体会一下它们之间的继承关系。

例 14.06 创建一个水果父类，在另一个苹果类中通过 extends 关键字来继承水果类中的成员属性和方法，最后对子类进行实例化操作。代码如下。**（实例位置：光盘\mr\14\sl\14.06）**

```
<?php
    class Fruit{
        var $apple="苹果";                  //定义变量
        var $banana="香蕉";
        var $orange="橘子";
    }
    class FruitType extends Fruit{          //类之间继承
        var $grape="葡萄";                  //定义子类变量
    }
    $fruit=new FruitType();                 //实例化对象
    echo "水果包含：".$fruit->apple."，".$fruit->banana."，".$fruit->orange."，".$fruit->grape;
?>
```

运行结果如图 14.9 所示。



图 14.9 类的继承

14.5.2 类的继承——parent::关键字

通过 parent::关键字也可以在子类中调用父类中的成员方法，其语法格式如下：

```
parents:: 父类的成员方法(参数);
```

下面通过 parent::关键字重新设计例 14.06 中的继承方法。在子类的 AppleFruit_Type()方法中，直接通过 parent::关键字调用父类中的 FruitType()方法。关键代码如下：

```
<?php
    class Fruit{                            //定义水果类
        var $apple="苹果";                  //定义变量
        var $banana="香蕉";
        var $orange="橘子";
        public function say(){              //定义 say 方法
            echo "，".$this->apple."，";    //利用 this 关键字输出本类中的变量
            echo $this->banana."，";
            echo $this->orange;
        }
    }
    class FruitType extends Fruit{          //类之间继承
        var $grape="葡萄";                  //定义子类变量
        public function show(){             //定义 show 方法
            parent::say();                  //利用关键字 parent 调用父类中的 say 方法
        }
    }
```

```
    $fruit=new FruitType();                    //实例化对象
    echo $fruit->grape;                        //调用子类变量
    $fruit->show();                            //调用子类 show 方法
?>
```

此时它的输出结果与例 14.06 是相同的。

14.5.3　覆盖父类方法

所谓覆盖父类方法，也就是使用子类中的方法将从父类中继承的方法进行替换，也叫方法的重写。覆盖父类方法的关键就是在子类中创建与父类中相同的方法，包括方法名称、参数和返回值类型。

例如，可以在子类中创建一个与父类方法同名的方法，这样就实现了方法的重写。关键代码如下：

```
<?php
    class Car{                                 //定义轿车类
        protected $wheel;                      //定义保护变量
        protected $steer;
        protected $speed;
        public function say_type(){            //定义轿车类型方法
            $this->wheel="45.9cm";             //定义车轮直径长度
            $this->steer="15.7cm";             //定义方向盘直径长度
            $this->speed="120m/s";             //定义车速
        }
    }
    class SmallCar extends Car{                //定义小型轿车类继承轿车类
        public function say_type_Q7(){         //定义 Q7 轿车类型
            $this->wheel="50.9cm";             //定义车轮直径长度
            $this->steer="20cm";               //定义方向盘直径长度
            $this->speed="160m/s";             //定义车速
        }
        public function say_show(){            //定义输出方法
            $this->say_type_Q7();              //调用本类中方法
            echo "Q7 轿车轮胎尺寸：".$this->wheel."<br>";      //输出本类中定义的车轮直径长度
            echo "Q7 轿车方向盘尺寸：".$this->steer."<br>";    //输出本类中定义方向盘直径长度
            echo "Q7 轿车最高时速：".$this->speed;             //输出本类中定义的最高时速
        }
    }
    $car=new SmallCar();                                       //实例化小轿车类
    $car->say_show();                                          //调用 say_show()方法
?>
```

运行结果如图 14.10 所示。



图 14.10　重写方法

提示： 当父类和子类中都定义了构造方法时，子类的对象被创建后，将调用子类的构造方法，而不会调用父类的构造方法。

14.6 抽象类和接口

抽象类（Abstract）和接口（Interface）都是不能被实例化的特殊类。它们都是配合面向对象的多态性一起使用。下面讲解它们的声明和使用方法。

14.6.1 抽象类

抽象类是一种不能被实例化的类，只能作为其他类的父类来使用。抽象类使用 abstract 关键字来声明，其语法格式如下：

```
abstract class 抽象类名称{
    //抽象类的成员变量列表
    abstract function 成员方法 1( 参数 );          //定义抽象方法
    abstract function 成员方法 2( 参数 );          //定义成员方法
}
```

抽象类和普通类相似，包含成员变量、成员方法。两者的区别在于，抽象类至少要包含一个抽象方法。抽象方法没有方法体，其功能的实现只能在子类中完成。抽象方法也是使用 abstract 关键字来修饰。

注意： *在抽象方法后面要有分号“;”。*

抽象类和抽象方法主要应用于复杂的层次关系中，这种层次关系要求每一个子类都包含并重写某些特定的方法。

例如，中国的美食是多种多样的，有吉菜、鲁菜、川菜、粤菜等。每种菜系使用的都是煎、炒、烹、炸等手法，只是在具体的步骤上，各有各的不同。如果把中国美食当作一个大类 Cate，下面的各大菜系就是 Cate 的子类，而煎、炒、烹、炸则是每个类中都有的方法。每个方法在子类中的实现都是不同的，在父类中无法规定。为了统一规范，不同子类的方法要有一个相同的方法名，即 decoct（煎）、stir_fry（炒）、cook（烹）和 fry（炸）。

例 14.07 根据中国的美食，创建一个抽象类 Cate，在抽象类中定义 4 个抽象方法，即 decocts（煎）、stir_frys（炒）、cooks（烹）和 frys（炸）。创建吉、鲁、川、粤 4 个菜系子类，继承 Cate 类，并在子类中定义抽象方法，即 decocts（煎）、stir_frys（炒）、cooks（烹）和 frys（炸）。最后，实例化吉菜子类。关键代码如下。**（实例位置：光盘\mr\14\sl\14.07）**

```
<?php
    abstract class cate{                          //定义抽象类
        abstract function decocts();              //定义抽象方法煎
        abstract function stir_frys();            //定义抽象方法炒
        abstract function cooks();                //定义抽象方法烹
        abstract function frys();                 //定义抽象方法炸
    }
    class JL_Cate{                                //定义吉菜
        public function decocts($a,$b){           //定义煎方法
            echo "您点的菜是：".$a."<br>";        //输出菜名
            echo "价格是：".$b."<br>";            //输出价格
        }
        public function stir_frys($a,$b){         //定义炒方法
```

```
            echo "您点的菜是：".$a."<br>";          //输出菜名
            echo "价格是：".$b."<br>";              //输出价格
        }
        public function cooks($a,$b){              //定义烹方法
            echo "您点的菜是：".$a."<br>";          //输出菜名
            echo "价格是：".$b."<br>";              //输出价格
        }
        public function frys($a,$b){               //定义炸方法
            echo "您点的菜是：".$a."<br>";          //输出菜名
            echo "价格是：".$b."<br>";              //输出价格
        }
    }
//省略了部分代码
    $jl=new JL_Cate();                             //实例化吉菜系
    $jl->decocts("小鸡炖粉条","39 元");              //调用煎方法
?>
```

运行结果如图 14.11 所示。

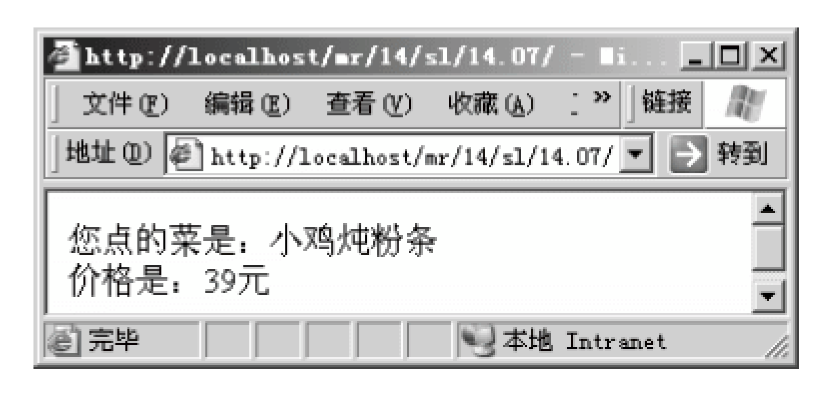


图 14.11　抽象类的应用

14.6.2　接口

继承特性简化了对象、类的创建，增加了代码的可重性。但 PHP 只支持单继承。如果想实现多重继承，就要使用接口。PHP 可以实现多个接口。

1．接口的声明

接口类通过 interface 关键字来声明，接口中声明的方法必须是抽象方法，接口中不能声明变量，只能使用 const 关键字声明为常量的成员属性，并且接口中所有成员都必须具备 public 的访问权限。接口声明的语法格式如下：

```
interface  接口名称{                              //使用 interface 关键字声明接口
    //常量成员                                    //接口中成员只能是常量
    //抽象方法                                    //成员方法必须是抽象方法
}
```

接口和抽象类相同都不能进行实例化的操作，也需要通过子类来实现。但是接口可以直接使用接口名称在接口外去获取常量成员的值。

例如，下面声明一个 One 接口，其代码如下：

```
interface One{                                   //声明接口
    const CONSTANT='CONSTANT value';             //声明常量成员属性
    function FunOne();                           //声明抽象方法
}
```

接口之间也可以实现继承，同样需要使用 extends 关键字。

例如，下面声明一个 Two 接口，通过 extends 关键字继承 One。代码如下：

```
interface Two extends One{                    //声明接口，并实现接口之间的继承
	function FunTwo();                        //声明抽象方法
}
```

2．接口的应用

因为接口不能进行实例化的操作，所以要使用接口中的成员，那么就必须借助子类。在子类中继承接口使用 implements 关键字。如果要实现多个接口的继承，那么每个接口之间使用逗号“,” 连接。

提示：既然通过子类继承了接口中的方法，那么接口中的所有方法必须都在子类中实现，否则 PHP 将抛出如图 14.12 所示的错误信息。

```
Fatal error: Class Member contains 2 abstract methods and
must therefore be declared abstract or implement the
remaining methods (Popedom::setPopedom,
Popedom::getPopedom) in F:\PkhPHP\www\MR\Instance\12\12.8
\index.php on line 27
```

图 14.12　接口中方法没有在子类中全部实现

下面看一个接口的实际应用，如例 14.08 所示。

例 14.08　在本例中，首先声明两个接口 Person 和 Popedom。然后在子类 Member 中继承接口并声明在接口中定义的方法。最后实例化子类，调用子类中的方法输出数据。代码如下。**（实例位置：光盘\mr\14\sl\14.08）**

```
<?php
	interface Person{                              //定义 Person 接口
		public function say();                     //定义接口方法
	}
	interface Popedom{                             //定义 Popedom 接口
		public function money();                   //定义接口方法
	}
	class Member implements Person,Popedom{        //类 Member 实现接口 Person 和 Propedom 接口
		public function say(){                     //定义 say 方法
			echo "我只是一名普通员工，";            //输出信息
		}
		public function money(){                   //定义方法 money
			echo "我一个月的薪水是 10000 元";       //输出信息
		}
	}
	$man=new Member ();                            //实例化对象
	$man->say();                                   //调用 say 方法
	$man->money();                                 //调用 money 方法
?>
```

运行结果如图 14.13 所示。

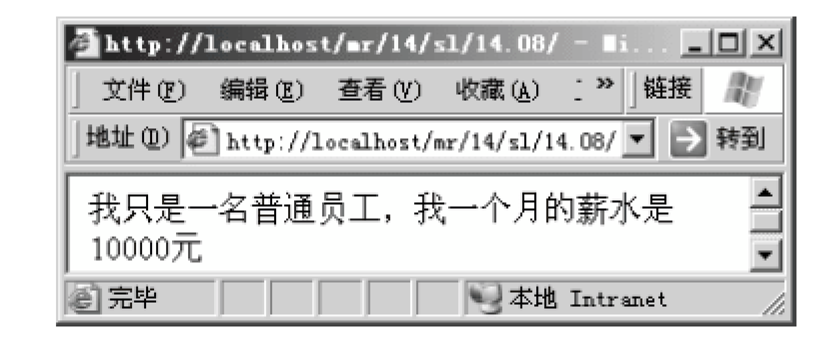
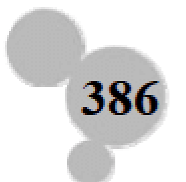


图 14.13　应用接口



14.7　面向对象的多态性

面向对象编程的特点之三是多态性，是指一段程序能够处理多种类型对象的能力。例如，在介绍面向对象特点时举的火车和汽车的例子，虽然火车和汽车都可以移动，但是它们的行为是不同的，火车要在铁轨上行驶，而汽车则在公路上行驶。

在 PHP 中，多态有两种实现方法，即通过继承实现多态和通过接口实现多态。

14.7.1　通过继承实现多态

继承性已经在前面讲解过，这里直接给出一个实例，展示通过继承实现多态的方法。

例 14.09　首先创建一个抽象类 Type，用于表示各种交通方法，然后让子类继承这个 Type 类。代码如下。**（实例位置：光盘\mr\14\sl\14.09）**

```
<?php
    abstract class Type{                              //定义抽象类 Type
        abstract function go_Type();                  //定义抽象方法 go_Type()
    }
    class Type_car extends Type{                      //小轿车类继承 Type 抽象类
        public function go_Type(){                    //重写抽象方法
            echo "我开着小轿车去拉萨";                //输出信息
        }
    }
    class Type_bus extends Type{                      //定义巴士车继承 Type 类
        public function go_Type(){                    //重写抽象方法
            echo "我坐巴士去拉萨";
        }
    }
    function change($obj){                            //自定义方法根据传入对象不同调用不同类中的方法
        if($obj instanceof Type){
            $obj->go_Type();
        }else{
            echo "传入的参数不是一个对象";            //输出信息
        }
    }
    echo "实例化 Type_car：";
    change(new Type_car());                           //实例化 Type_car 类
    echo "<br>";
    echo "实例化 Type_bus：";
    change(new Type_bus);                             //实例化 Type_bus 类
?>
```

运行结果如图 14.14 所示。

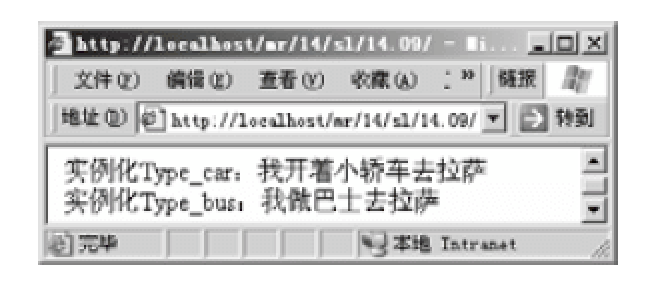


图 14.14　通过继承实现多态

在上述实例中对于抽象类 Type 而言，Car 类和 Bus 类就是其多态性的体现。

14.7.2 通过接口实现多态

下面通过实例讲解如何通过接口实现多态。

例 14.10 在本例中，首先定义接口，并定义一个空方法 go_Type()。然后定义 Type_car 和 Type_Bus 子类继承接口 Type。最后通过 instanceof 关键字检查对象是否属于接口 Type。代码如下。（**实例位置：光盘\mr\14\sl\14.10**）

```
<?php
	interface Type{								//定义 Type 接口
		public function go_Type();					//定义接口方法
	}
	class Type_car implements Type{				//Type_car 类实现 Type 接口
		public function go_Type(){					//定义 go_Type 方法
			echo "我开着小轿车去拉萨";				//输出信息
		}
	}
	class Type_bus implements Type{				//Type_bus 实现 Type 方法
		public function go_Type(){					//定义 go_Type 方法
			echo "我坐巴士去拉萨";					//输出信息
		}
	}
	function change($obj){						//自定义方法
		if($obj instanceof Type){
			$obj->go_Type();
		}else{
			echo "传入的参数不是一个对象";			//输出信息
		}
	}
	echo "实例化 Type_car：";
	change(new Type_car);						//实例化对象
	echo "<br>";
	echo "实例化 Type_bus：";
	change(new Type_bus);
?>
```

其运行结果与例 14.09 是相同的。

14.8 面向对象的关键字

14.8.1 final 关键字

final，中文含义是最终的、最后的。被 final 修饰过的类和方法就是“最终的版本”。如果有一个类的格式为：

```
final class class_name{
//…
}
```



说明该类不可以再被继承，也不能再有子类。

如果有一个方法的格式为：

```
final function method_name()
```

说明该方法在子类中不可以进行重写，也不可以被覆盖。

这就是 final 关键字的作用。

14.8.2　static 关键字——声明静态类成员

在 PHP 中，通过 static 关键字修饰的成员属性和成员方法被称为静态属性和静态方法。静态属性和静态方法不需要在被类实例化的情况下就可以直接使用。

1．静态属性

静态属性就是使用关键字 static 修饰的成员属性，它属于类本身而不属于类的任何实例。它相当于存储在类中的全局变量，可以在任何位置通过类来访问。静态属性访问的语法如下：

```
类名称::$静态属性名称
```

其中的符号“::”被称为范围解析操作符，用于访问静态成员、静态方法和常量，还可以用于覆盖类中的成员和方法。

如果要在类内部的成员方法中访问静态属性，那么在静态属性的名称前加上操作符“self::”即可。

2．静态方法

静态方法就是通过关键字 static 修改的成员方法。由于它不受任何对象的限制，所以可以不通过类的实例化直接引用类中的静态方法。静态方法引用的语法如下：

```
类名称::静态方法名称([参数 1,参数 2,……])
```

同样如果要在类内部的成员方法中引用静态方法，那么也是在静态方法的名称前加上操作符“self::”。

> 注意：在静态方法中，只能调用静态变量，而不能调用普通变量，普通方法则可以调用静态变量。

使用静态成员，除了可以不需要实例化对象，另一个作用就是在对象被销毁后，仍然保存被修改的静态数据，以便下次继续使用。

例 14.11　首先，声明一个静态变量$num，声明一个方法，在方法的内部调用静态变量并给变量值加 1。然后，实例化类中的对象。最后，调用类中的方法。代码如下。**（实例位置：光盘\mr\14\sl\14.11）**

```
<?php
	class Web{
		static $num="0";                              //定义静态变量
		public function change(){                     //定义 change 方法
			echo "您是本站第".self::$num."位访客.\t";     //输出静态变量信息
			self::$num++;                             //静态变量作自增运算
		}
	}
	$web=new Web();                                   //实例化对象
	echo "第一次实例化调用：<br>";
	$web->change();                                   //对象调用
	$web->change();
	$web->change();
	echo "<br>第二次实例化调用<br>";
```

```
        $web_wap=new Web();                                 //改变对象句柄实例化对象
        $web_wap->change();
        $web_wap->change();
?>
```

运行结果如图 14.15 所示。

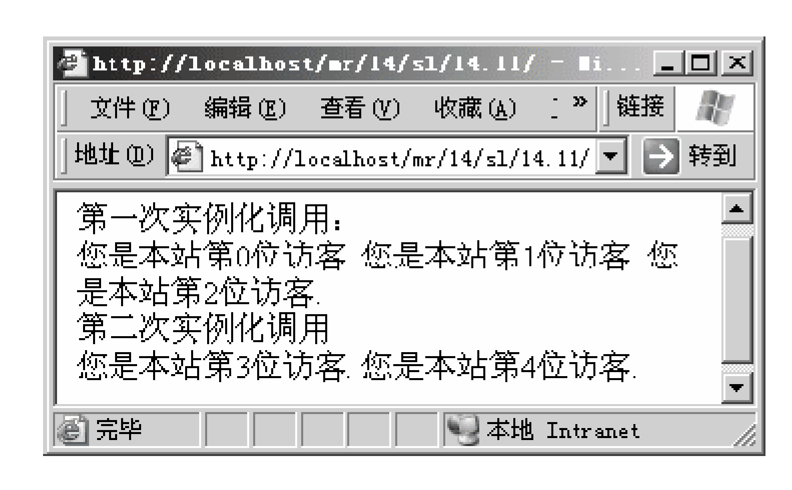


图 14.15　静态变量的使用

如果将程序代码中的静态变量改为普通变量，如“private $num = 0;”，那么结果就不一样了。读者可以动手试一试。

说明：静态成员不用实例化对象，当类第一次被加载时就已经分配了内存空间，所以直接调用静态成员的速度要快一些。但如果静态成员声明得过多，空间一直被占用，反而会影响系统的功能。这个尺度只能通过实践积累才能真正地把握。

14.8.3　clone 关键字——克隆对象

1．克隆对象

对象的克隆可以通过关键字 clone 来实现。使用 clone 克隆的对象与原对象没有任何关系，它是将原对象从当前位置重新复制了一份，也就是相当于在内存中新开辟了一块空间。clone 关键字克隆对象的语法格式如下：

```
$克隆对象名称=clone $原对象名称;
```

对象克隆成功后，它们中的成员方法、属性以及值是完全相同的。如果要为克隆后的副本对象在克隆时重新为成员属性赋初始值，那么就要使用到下面将要介绍的魔术方法“__clone()”。

2．克隆副本对象的初始化

魔术方法“__clone()”可以为克隆后的副本对象重新初始化。它不需要任何参数，其中自动包含$this 和$that 两个对象的引用，$this 是副本对象的引用，$that 则是原本对象的引用。

例 14.12　本例中，在对象$book1 中创建__clone()方法，将变量$object_type 的默认值从 book 修改为 computer。使用对象$book1 克隆出对象$book2，输出$book1 和$book2 的$object_type 值。代码如下。（**实例位置：光盘\mr\14\sl\14.12**）

```
<?php
class Book{                                         //类 Book
        private $object_type = 'book';              //声明私有变量$object_type，并赋初值为 book
        public function setType($type){             //声明成员方法 setType，为变量$object_type 赋值
                $this -> object_type = $type;
        }
        public function getType(){                  //声明成员方法 getType，返回变量$object_type 的值
                return $this -> object_type;
```

```
        }
        public function __clone(){                              //声明__clone()方法
                $this ->object_type = 'computer';               //将变量$object_type 的值修改为 computer
        }
}
$book1 = new Book();                                            //实例化对象$book1
$book2 = clone $book1;                                          //使用普通数据类型的方法给对象$book2 赋值
echo '对象$book1 的变量值为：'.$book1 -> getType();              //输出对象$book1 的值
echo '<br>';
echo '对象$book2 的变量值为：'.$book2 -> getType();
?>
```

运行结果如图 14.16 所示。

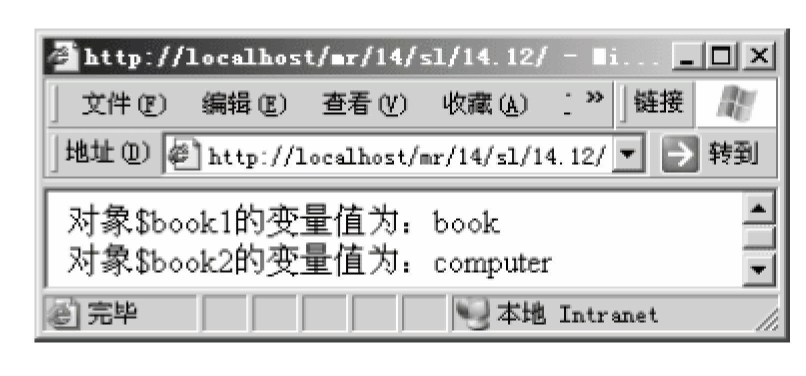


图 14.16　__clone()方法

对象$book2 克隆了对象$book1 的全部行为及属性，而且还拥有属于自己的成员变量值。

14.9　面向对象的魔术方法

PHP 中有很多以两个下划线开头的方法，如前面已经介绍过的__construct()、__destruct()和__clone()，这些方法被称为魔术方法。

14.9.1　__set()和__get()方法

__set()和__get()方法对私有成员进行赋值或者获取值的操作。

- ☑ __set()方法：在程序运行过程中为私有的成员属性设置值，它不需要任何返回值。__set()方法包含两个参数，分别表示变量名称和变量值。两个参数不可省略。这个方法不需要主动调用，可以在方法前加上 private 关键字修饰，防止用户直接调用。
- ☑ __get()方法：在程序运行过程中，在对象的外部获取私有成员属性的值。它有一个必要参数，即私有成员属性名，它返回一个允许对象在外部使用的值。这个方法同样不需要主动调用，可以在方法前加上 private 关键字，防止用户直接调用。

14.9.2　__isset()和__unset()方法

__isset()和__unset()方法如果不看它们前面的“__”符号，我们一定会想到 isset()和 unset()函数。

isset()函数用于检测变量是否存在，如果存在则返回 TRUE，否则返回 FALSE。而在面向对象中，通过 isset()函数可以对公有的成员属性进行检测，但是对于私有的成员属性，这个函数就不起作用了，而魔术方法__isset()的作用就是帮助 isset()函数检测私有成员属性。

如果在对象中存在“__isset()”方法，当在类的外部使用 isset()函数检测对象中的私有成员属性时，就

会自动调用类中的“__isset()”方法完成对私有成员属性的检测操作。其语法格式如下：

```
bool_isset(string name)        //传入对象中的成员属性名，返回值为测定结果
```

unset()函数的作用是删除指定的变量，参数为要删除的变量名称。而在面向对象中，通过 unset()函数可以对公有的成员属性进行删除操作，但是对于私有的成员属性，那么就必须有__unset()方法的帮助才能够完成。

__unset()方法帮助 unset()函数在类的外部删除指定的私有成员属性。其语法格式如下：

```
void_unset(string name)        //传入对象中的成员属性名，执行将私有成员属性删除的操作
```

14.9.3 __call()方法

__call()方法的作用是，当程序试图调用不存在或不可见的成员方法时，PHP 会先调用__call()方法来存储方法名及其参数。__call()方法包含两个参数，即方法名和方法参数。其中，方法参数是以数组形式存在的。

例 14.13 本例中声明一个类 MrSoft，包含两个方法，即 MingRi()和__call()。类实例化后，调用一个不存在的方法 MingR()，看魔术方法__call()的妙用。代码如下。**（实例位置：光盘\mr\14\sl\14.13）**

```
<?php
class MrSoft{
    public function MingRi(){                                   //方法 MingRi()
        echo '调用的方法存在，直接执行此方法。<p>';
    }
    public function __call($method, $parameter) {               //__call()方法
        echo '如果方法不存在，则执行__call()方法。<br>';
        echo '方法名为：'.$method.'<br>';                        //输出第一个参数，即方法名
        echo '参数有：';
        var_dump($parameter);                                   //输出第二个参数，是一个参数数组
    }
}
$mrsoft = new MrSoft();                                         //实例化对象$mrsoft
$mrsoft -> MingRi();                                            //调用存在的方法 MingRi()
$mrsoft -> MingR('how','what','why');                           //调用不存在的方法 MingR()
?>
```

运行结果如图 14.17 所示。

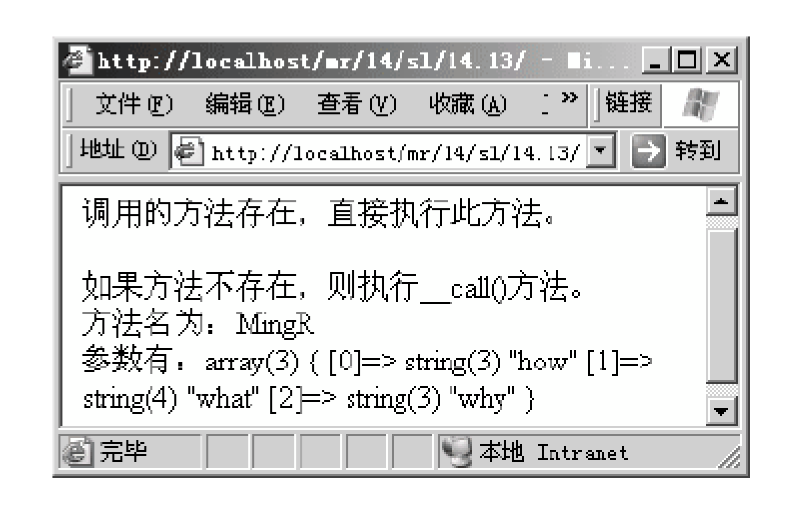


图 14.17 __call()方法

14.9.4 __toString()方法

魔术方法__toString()的作用是，当使用 echo 或 print 输出对象时，将对象转化为字符串。

例 14.14 本例中，定义 People 类，应用__toString()方法输出 People 类的实例化对象$peo。代码如下。



（实例位置：光盘\mr\14\sl\14.14）

```
<?php
    class People{
        public function __toString(){
            return "我是 toString 的方法体";
        }
    }
    $peo=new People();
    echo $peo;
?>
```

运行结果为：我是 toString 的方法体。

说明：（1）如果没有__toString()方法，直接输出对象将会发生致命错误（fatal error）。

（2）输出对象时应注意，echo 或 print 函数后面直接跟要输出的对象，中间不要加多余的字符。否则__toString 方法不会被执行。如 echo '字串'.$myComputer、echo ' '.$myComputer 等都不可以，一定要注意。

14.9.5 __autoload()方法

将一个独立、完整的类保存到一个 PHP 页中，并且文件名和类名保持一致，这是每个开发人员都需要养成的良好习惯。这样，在下次重复使用某个类时就可以很轻松地找到它。但还有一个问题是让开发人员头疼不已的，如果要在一个页面中引进很多的类，需要使用 include_once()函数或 require_once()函数一个一个地引入。

在 PHP 5 中应用__autoload()方法解决了这个问题。__autoload()方法可以自动实例化需要使用的类。当程序要用到一个类，但该类还没有被实例化时，PHP 5 将使用__autoload()方法，在指定的路径下自动查找和该类名称相同的文件。如果找到则继续执行，否则报告错误。

例 14.15 首先创建一个类文件 inc.php，该文件包含类 People。然后创建 index.php 文件，在文件中创建__autolaod()方法，判断类文件是否存在，如果存在则使用 require_once()函数将文件动态引入，否则输出提示信息。类文件 inc.php 的代码如下。**（实例位置：光盘\mr\14\sl\14.15）**

```
<?php
    class People{                                   //定义类
        public function __toString(){               //定义__toString 方法
            return"自动加载类";
        }
    }
?>
```

index.php 文件的代码如下：

```
<?php
    function __autoload($class_name){               //创建__autoload()方法
        $class_path = $class_name.'/inc.php';       //类文件路径
        if(file_exists($class_path)){               //判断类文件是否存在
            include_once($class_path);
                //动态包含类文件
        }else
            echo '类路径错误。';
    }
```

```
        $mrsoft = new People();                                          //实例化对象
        echo $mrsoft;                                                    //输出类内容
?>
```

运行结果为：自动加载类

14.10 照猫画虎——基本功训练

14.10.1 基本功训练 1——编写一个矩形类

视频讲解：光盘\mr\14\lx\编写一个矩形类.exe

实例位置：光盘\mr\14\zmhh\01

在本实例中定义了长度、宽度、长度单位、面积单位等。其实现思想是通过构造函数动态地将长度和宽度等传递到类中的构造函数中，实现面积的计算。运行结果如图 14.18 所示。



图 14.18 矩形类数据输出

操作步骤如下：

（1）声明一个类文件，定义 3 个成员属性。其中包括 width（长度）、height（高度）和单位 unit。

（2）定义构造方法__construct()将 3 个成员属性指向自定义传递的 3 个属性值。

（3）实例化对象并实现输出效果。代码如下：

```
<?php
        header("Content-Type:text/html;charset=utf-8");                  //设置编码集
        class Rect {                                                     //声明类
                var $width;                                              //定义成员属性
                var $height;
                var $unit;
                function __construct($width,$height,$unit) {             //构造方法
                        $this->width=$width;
                        $this->height=$height;
                        $this->unit=$unit;
                        $this->area();
                }
                function area(){
                        echo "矩形的宽为：".$this->width.$this->unit."<br>";
                        echo "矩形的长为：".$this->height.$this->unit."<br>";
                        echo "矩形的面积为：".($this->height*$this->width).$this->unit."<sup>2</sup>";
                }
        }
        $re=new Rect("10","20","cm");                                    //实例化对象
?>
```

照猫画虎：编写一个矩形体类。（20 分）（实例位置：光盘\mr\14\zmhh\01_zmhh）

14.10.2　基本功训练 2——用构造方法初始化

视频讲解：光盘\mr\14\lx\用构造方法初始化.exe

实例位置：光盘\mr\14\zmhh\02

类中很重要的一个方法是构造方法，它的主要作用是当类被实例化的同时调用相关的方法或属性。如果在类中用户没有定义构造方法，那么程序将自动定义一个什么都不做的构造方法。运行本实例，结果如图 14.19 所示。

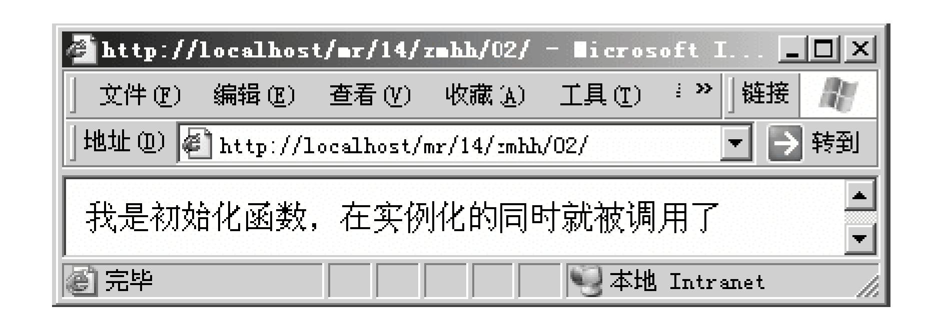


图 14.19　构造方法初始化

实现本实例原理非常简单，利用__construct 关键字定义构造函数，并接受类实例化的同时传递进来的参数。然后再进行相关操作。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");			//定义编码集风格
	class Class_name {							//定义类
		function __construct() {					//定义构造方法
			echo "我是初始化函数，在实例化的同时就被调用了";
		}
	}
	new Class_name();
?>
```

照猫画虎：通过 PHP 4 的方式编写构造方法。（20 分）（实例位置：光盘\mr\14\zmhh\02_zmhh）

14.10.3　基本功训练 3——this 关键字的用途

视频讲解：光盘\mr\14\lx\this 关键字的用途.exe

实例位置：光盘\mr\14\zmhh\03

this 关键字在任何一个面向对象的编程语言当中都存在它的身影，此关键字的主要作用是调用本类中的成员属性和成员方法。需要注意的是，此关键字只能调用本类中的抽象方法。运行本实例，结果如图 14.20 所示。

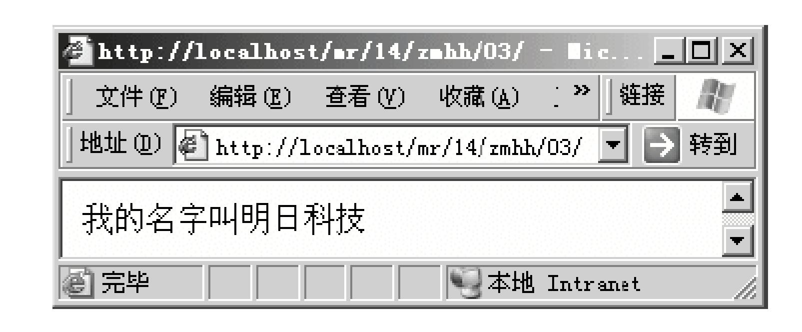


图 14.20　this 关键字的使用

本实例主要是 this 关键字的强化训练，其实 this 关键字的使用非常简单，用户只需记住此关键字只应用

于本类中的成员属性和方法即可。本实例的代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");          //设置编码集风格
    class This{                                              //声明类
        var $name;                                           //定义成员属性
        function __construct($name){                         //初始化对象
            $this->name=$name;                               //调用本类中的成员属性
            $this->say();                                    //调用本类中的成员方法
        }
        function say(){
            echo "我的名字叫".$this->name;
        }
    }
    new This("明日科技");                                    //实例化对象
?>
```

照猫画虎： 利用 final 关键字声明类，通过继承并实例化子类观察显示结果。（20 分）（实例位置：光盘\mr\14\zmhh\03_zmhh）

14.10.4 基本功训练 4——方法参数传递

视频讲解：光盘\mr\14\lx\方法参数传递.exe

实例位置：光盘\mr\14\zmhh\04

用户在定义成员方法时很多时候是需要传递参数的。传递参数分为两种情况：第一种情况是按值传递，即直接传递的是数据。第二种情况是按址传递，按址传递不同于按值传递，它传递进来的是一个引用地址或者说是一个对象。本实例采用按值传递的方式传递数据，其运行结果如图 14.21 所示。

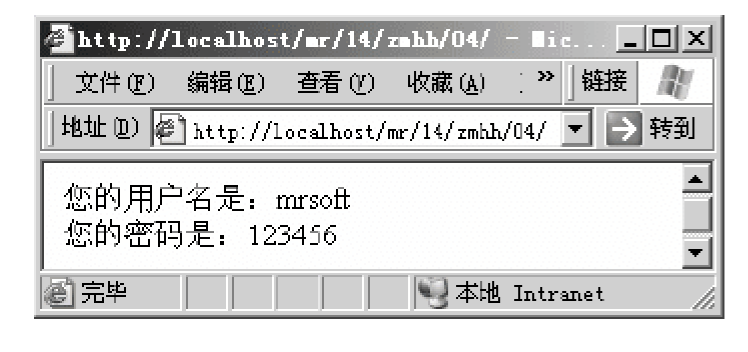


图 14.21 方法参数的传递

操作步骤如下：

（1）声明类文件并命名为 Del，定义两个成员属性和一个成员方法，此方法接收两个参数，这两个参数利用 this 关键字将传递进来的数据指向成员属性。

（2）实例化对象，通过对象句柄调用成员方法并传递参数。本实例的程序代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");
    class Del{
        var $name;
        var $pwd;
        function say($name,$pwd){
            $this->name=$name;
            $this->pwd=$pwd;
            echo "您的用户名是：".$this->name."<br>";
            echo "您的密码是：".$this->pwd;
        }
    }
    $del=new Del();
    $del->say("mrsoft","123456");
?>
```

照猫画虎：实现对象的传递。（20 分）（实例位置：光盘\mr\14\zmhh\04_zmhh）

14.10.5　基本功训练 5——方法的覆盖

视频讲解：光盘\mr\14\lx\方法的覆盖.exe
实例位置：光盘\mr\14\zmhh\05

方法的覆盖是 PHP 面向对象开发中很重要的一个知识点，一般多出现在类的继承和接口的实现之中。运行本实例，结果如图 14.22 所示。

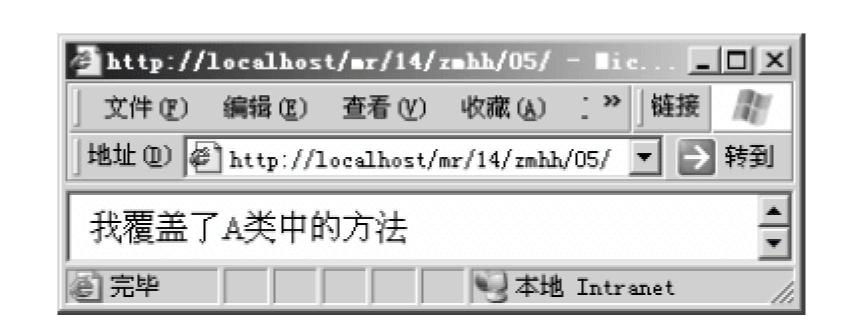


图 14.22　方法的覆盖

本实例采用的是类之间的继承关系，将 B 类继承自 A 类并覆盖父类（A 类中的）say()方法。实现方法的覆盖必须确定覆盖与被覆盖的方法名称相同。本实例的代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");			//设置编码集风格
	class A{									//声明类 A
		function say(){							//定义 say 方法
			echo "我 A 类中的方法";
		}
	}
	class B extends A{							//让 B 类继承自 A 类
		function say(){							//覆盖父类中的 say 方法
			echo "我覆盖了 A 类中的方法";
		}
	}
	$b=new B();								//实例化子类
	$b->say();								//调用子类中的 say 方法
?>
```

照猫画虎：方法的重载。（20 分）（实例位置：光盘\mr\14\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

14.11　情景应用——拓展与实践

14.11.1　情景应用 1——数据统计类中定义求数值平方的方法

视频讲解：光盘\mr\14\lx\数据统计类中定义求数值平方的方法.exe
实例位置：光盘\mr\14\qjyy\01

运行本实例，如图 14.23 所示。首先在图中文本框中输入一组数字并用半角逗号分割，然后单击“计算”按钮将在页面中打印出该组数字的平均值。

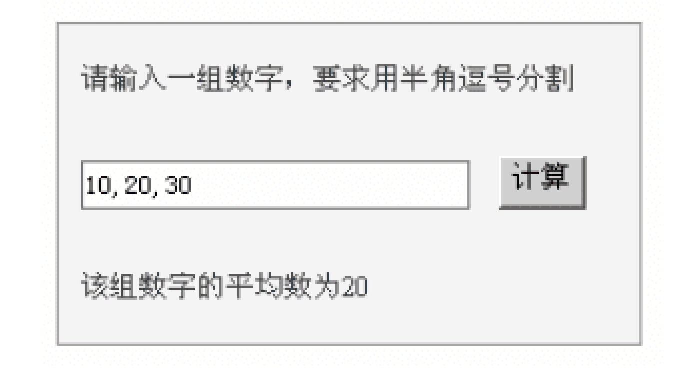


图 14.23　计算一组数的平均值

实现本实例时，由于该类只包含一个用于计算平均数的方法，所以在实现该类时将求平均数的方法定义为静态方法，这样在不实现该类的情况下通过类名即可直接调用该方法。

在 PHP 面向对象编程方式中，静态方法隶属于某个类，但不受类的束缚，可以直接被外部访问。与定义类中的普通方法相比，定义静态方法时需要使用 static 关键字进行修饰，其定义格式如下：

```
public static function  方法名()
{
    //该方法实现的功能
}
```

在类体内部调用该方法时，使用 self 关键字加“::”调用，在类外直接使用类名加“::”引用。

（1）建立数据统计类 Stat，并在类体内定义静态方法 getAvg()用来实现求多个数字的平均值，其实现代码如下：

```
class Stat
{
    public static function getAvg ($arryNum)            //定义静态方法来求平均值
    {
        $totalNum = count($arryNum);                    //获得数字总数
        if ($totalNum == 0) {                           //如果数字个数为 0 则返回 null
            return null;
        } else {
            $sum = 0;                                   //用变量$sum 保存所有数字的和
            for ($i = 0; $i < $totalNum; $i ++)          //通过循环计算所有数字的和
                $sum += $arryNum[$i];
            }
            return $sum / $totalNum;                    //返回平均数
        }
    }
}
```

上述代码用于实现求平均数的 getAvg()方法中，为了能够实现求任意个数数字的平均数，将数组作为参数，这样只要求得该数组所有元素的平均值即可。在具体实现过程中，首先使用函数 count()计算数组元素的个数，然后使用 for 循环语句计算数组中所有元素的和，最后使用 return 语句返回所求得的平均数。

（2）建立数字录入表单，当用户单击表单中的“计算”按钮时，将通过如下代码计算在表单中所录入数字的平均数。

```
if(isset($_POST['nums']) && trim($_POST['nums'])!=''){          //判断是否已经提交表单
    require 'Stat.php';                                         //包含数据统计类
    $arrayNum = explode(',', $_POST['nums']);                   //将提交数字保存到数组中
    echo '该组数字的平均数为' . Stat::getAvg($arrayNum);        //调用统计类的静态方法 getAvg()求平均数
}
```

上述代码首先判断用户是否提交了表单，如果是则首先使用 require 语句包含数据统计类，然后使用函



数 explode()用半角逗号分割所提交数字字符串，最后调用数据统计类的静态方法 getAvg()计算所录入数字的平均数。

DIY：使用重载实现不同类型数据的运算。提示：应用面向对象编程中的方法重载实现。（20 分）（实例位置：光盘\mr\14\qjyy\01_diy）

14.11.2　情景应用 2——封装一个数据库连接类

视频讲解：光盘\mr\14\lx\封装一个数据库连接类.exe
实例位置：光盘\mr\14\qjyy\02

在 PHP 的 Web 程序开发中，连接 MySQL 数据库是一件非常平常的事情。无论是开发一个什么样的 Web 程序，都会涉及数据库的操作。如果用户习惯性使用面向过程的思维方式，其实也是未尝不可的事情。但是不得不承认，通过面向对象的方式开发的程序无论是系统执行效率还是代码的思维逻辑都要优于面向过程的开发方式。运行本实例，结果如图 14.24 所示。

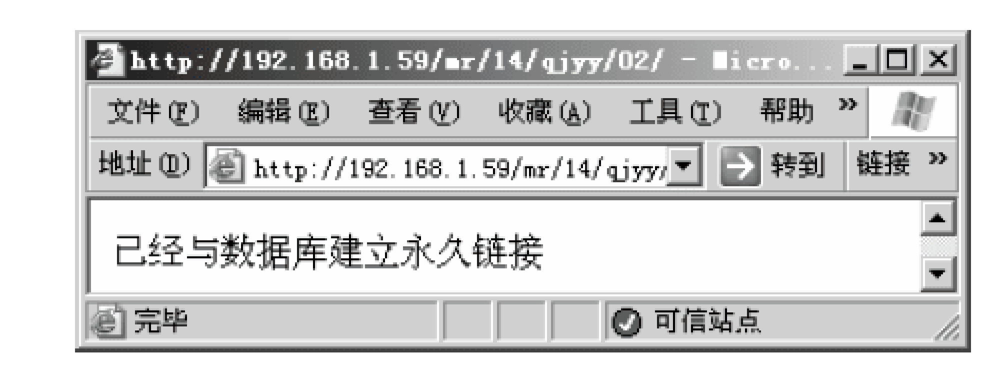


图 14.24　封装一个数据库连接类

操作步骤如下：

（1）新建脚本文件 index.php。结合类命名格式定义类名为 Mysql。

（2）定义 4 个私有变量，这 4 个变量中前 3 个参数是连接 MySQL 数据库必需的参数，第 4 个参数是作为连接句柄出现的。

（3） 定义构造函数，将外部传入的参数变量转换为类内部变量并要求本类在实例化的同时调用 connect()方法。

（4）编写 connect()方法，通过 mysql_connect()和 mysql_pconnect()函数连接 MySQL 服务器。

（5）最后，实例化对象，传递数据库连接参数（服务器、用户名、密码和连接方式）。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");
/*
	 命名规则:
		（1）类的命名规则，类名称单词首字母大写。
		（2）方法名称首字母小写，如果方法名称是由多个单词组成，那么从第二个单词首字母大写。
		（3）变量名称首字母一般小写。
		（4）常量名称一般全部大写。
*/
	class Mysql{                                    //创建名称为 Mysql 的类
		private $localhost;                         //定义连接数据库相关的私有变量数据源
		private $username;                          //私有变量数据库用户名
		private $pwd;                               //私有变量数据库密码
		private $conn;                              //私有变量连接符
		public function __construct($localhost,$username, $pwd,$conn){          //定义构造函数
			$this->localhost=$localhost;
			$this->username=$username;
			$this->pwd=$pwd;
			$this->conn=$conn;
			$this->connect();
```

```
        }
        public function connect(){                              //当实例化对象时调用 connect()方法
            if($this->conn=="pconn"){                           //当$conn 变量为 pconn 时
                //建立永久连接
                if($this->conn=mysql_pconnect($this->localhost,$this->username,$this->pwd)){
                    echo "已经与数据库建立永久连接";              //输出信息
                }else{
                    echo "连接数据库出现错误";
                }
            }else{
                //建立暂时连接
                if($this->conn=mysql_connect($this->localhost,$this->username,$this->pwd)){
                    echo "已经与数据库建立连接";
                }else{
                    echo "连接数据库出现错误";
                }
            }
        }
    }
    $mysql=new mysql("127.0.0.1","root","111","pconn");        //实例化对象
?>
```

DIY：封装 PDO 连接类。（20 分）（实例位置：光盘\mr\14\qjyy\02_diy）

14.11.3 情景应用 3——封装一个数据库操作类

视频讲解：光盘\mr\14\lx\封装一个数据库操作类.exe

实例位置：光盘\mr\14\qjyy\03

连接数据库的目的是为了操作数据库，操作数据库无外乎增、删、改、查这几项内容，用户可以把操作数据库的相关内容同样封装到类中，可能对于面向对象知识薄弱的用户来说做这样的事情是有些难度的，但是在本项目中为用户封装了一个数据库操作类，读者可以通过这个封装类进行学习，开发出属于自己的数据库操作类。本实例的运行结果如图 14.25 所示。

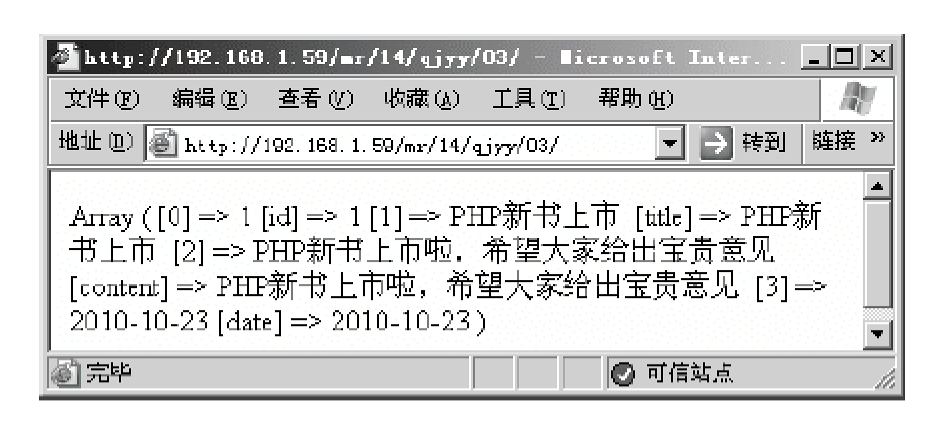


图 14.25 封装数据库的操作类

操作步骤如下：

（1）封装 AdminDB 类，定义 executeSQL()方法，在该方法中通过 substr()函数截取 SQL 语句中的前 6 个字节，通过这 6 个字节判断 SQL 语句的类型，通过 mysql_query()函数执行 SQL 语句。

如果 SQL 语句的类型是 select，则通过 mysql_fetch_array()函数获取结果集，如果结果集为真则返回结果集，如果无结果集则返回 FALSE；如果 SQL 的类型是 update、delete 或者 insert，直接返回 mysql_query()函数的返回值。AdminDB 类的完整代码如下：

```
class AdminDB{
function executeSQL ($sql, $connID){
    $sqlType = strtolower(substr(trim($sql), 0, 6));     //提取 SQL 语句的类型
    $rs = mysql_query($sql,$connID);                     //执行 SQL 语句
     if ($sqlType == 'select') {                         //如果是 select 查询
```

```
        $arrayData = mysql_fetch_array($rs);            //返回查询记录集
            if (count($arrayData) == 0 || $rs == false) {  //如果没查询到或发生错误
              return false;                             //返回 false
            } else {                                    //否则
                    return $arrayData;                  //返回记录集
            }
        } elseif ($sqlType == 'insert' || $sqlType == 'update' || $sqlType == 'delete') { //如执行插入、更新或删除语句
            return $rs;                                 //返回语句执行状态，即成功返回 TRUE，失败返回 FALSE
        } else {
            return false;                               //如果不是上述查询，则返回 FALSE
        }
    }
}
```

（2）实例化数据库连接类和数据库操作类，返回数据的查询结果。代码如下：

```
$connobj=new ConnDB("localhost","root","111","conn","utf8","db_database14");      //实例化数据库连接类
$conn=$connobj->connect();                                  //返回连接标识
$admindb=new AdminDB();                                     //数据库操作类实例化
$res=$admindb->executeSQL ("select * from tb_demo01",$conn);    //调用数据库操作类中的方法执行查询语句
if($res){                                                   //如果返回结果为真，则输出数据
      print_r($res);                                        //输出一个数组
}
```

DIY：扩展本操作类，利用 self 关键字调用静态方法。（20 分）（实例位置：光盘\mr\14\qjyy\03_diy）

14.11.4　情景应用 4——封装分页类

视频讲解：光盘\mr\14\lx\封装分页类.exe

实例位置：光盘\mr\14\qjyy\04

本实例相对本堂课的其他实例来说是比较有难度的。本实例主要讲解的是如何制作一个分页类。分页的主要功能是当页面数据过长或过大时，程序员可以根据需要将数据分为多页输出，这样做的好处是：一方面使页面简洁不会让用户有反感的情绪，另一方面在处理超长文本时缩短程序执行时间使程序运行更快捷。本实例运行结果如图 14.26 所示。

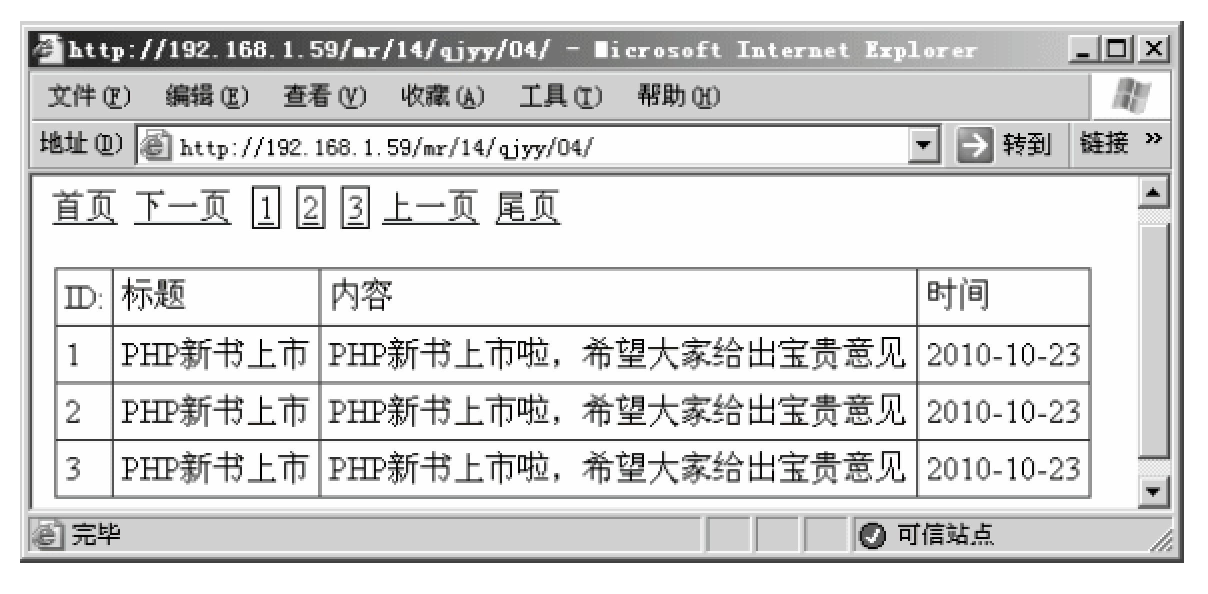


图 14.26　分页类

完成本实例，主要通过指定 SQL 语句的 limit 函数，要求页面输出指定的$pagesize 条数据，并利用拼接字符串的方法拼接地址栏参数，实现页面跳转的同时使数据分页输出。本实例同样利用了面向对象继承的相关知识，将数据库的连接相关方法定义到 conn.php 中，并在分页类中需要的地方进行实例化操作。代码如下：

```
<?php
	public function pageNow(){					//定义当前页
			if(isset($_GET[page])){				//如果页面中存在 page 参数
				$this->page=$_GET[page];		//定义类中变量 page 等于地址栏取得的参数
			}else{
				$this->page='1';			//否则类变量 page 等于 1
			}
			return $this->page;				//将变量 page 返回到方法调用处
		}
		public function number(){				//定义数据库中数据条数的方法
			new Mysql();					//实例化对象
			$rs=mysql_query("select * from $this->table");	//执行查询操作并返回结果集
			return $num=mysql_num_rows($rs);		//将数据库数据条数返回到方法调用处
		}
		public function pageFirst(){				//定义首页方法
?>
	//省略了部分代码
<?php
			}
		}
		public function pageCode(){				//定义指定页数跳转方法
			//定义 for 循环语句执行指定页数跳转
			for($a=0;$a<(ceil($this->number())/($this->pagesize));$a++){
?>
		<a style="border:1px solid #000000;margin:2px;padding:0px 2px;"href="index.php?page=<?php echo
($a+1);?>"><?php echo ($a+1);?></a>
<?php
			}
		}
?>
```

DIY：扩展本分页类，利用__toString()方法输出提示内容。（20 分）（实例位置：光盘\mr\14\qjyy\04_diy）

14.11.5　情景应用 5——封装特殊字符转义类

视频讲解：光盘\mr\14\lx\封装特殊字符转义类.exe

实例位置：光盘\mr\14\qjyy\05

特殊字符转义需要使用 addslashes()函数，一般常用于对 SQL 语句的转义。本实例利用面向对象的知识将 addslashes()函数封装到类中，当实例化对象后根据传递进来的参数进行特殊字符转义操作。本实例运行结果如图 14.27 所示。

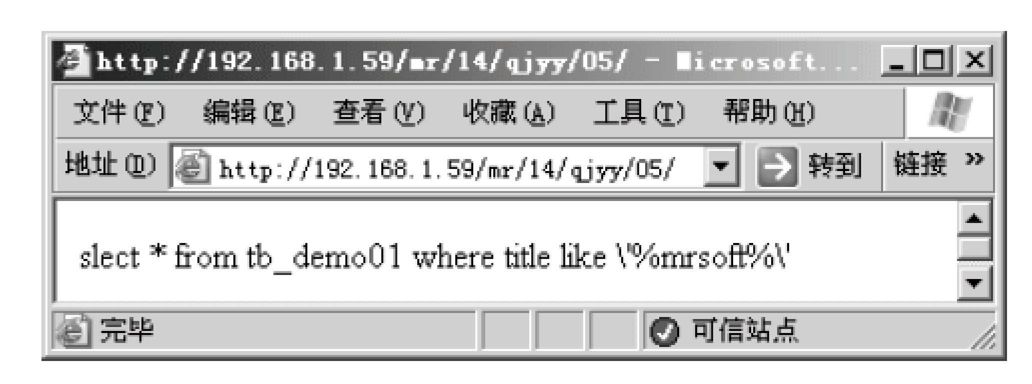


图 14.27　转义特殊字符

实现本实例原理非常简单，首先定义类名称为 Addslashes，然后定义成员方法 add()并接收一个外界参



数，当实例化对象后，利用对象句柄调用成员方法即可实现本操作。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");				//设置编码集风格
	class Addslashes{								//定义类名称
		function add($string){						//定义成员方法
			echo addslashes($string);
		}
	}
	$add=new Addslashes();							//实例化对象
	$string="slect * from tb_demo01 where title like '%mrsoft%'";
	$add->add($string);
?>
```

DIY：扩展本转义类，实现将转义后的字符串还原。（20 分）（实例位置：光盘\mr\14\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

14.12　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．实现类之间的继承需要使用关键字（　　）。

A．public　　B．set　　C．extends

2．构造方法需要使用关键字（　　）。

A．__construct　　B．__destruct　　C．__set　　D．__get

3．使类不能被其他子类所继承需要使用关键字（　　）。

A．private　　B．static　　C．final

4．声明接口需要使用关键字（　　）。

A．class　　B．abstract　　C．interface

5．PHP 中的类继承是下面哪一个（　　）。

A．多继承　　B．单继承　　C．既可以单继承也可以多继承

二、填空题（每题 10 分，5 道题）

1．将类 People 声明为抽象类的代码为（　　　　　　　　　　　　）。

2．克隆对象需要使用（　　　　　）关键字。

3．声明静态变量需要使用关键字（　　　　）。

4．检测某个对象是否从属于某个类需要使用关键字（　　　　）。

5．程序运行后，页面中显示的内容为（　　　）。

```
class Class_name {
		function __construct() {
			echo "我是初始化函数，在实例化的同时就被调用了";
```

```
        }
    }
    new Class_name();
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

14.13 行 动 指 南

开始日期：______年____月____日　　结束日期：______年____月____日

序号	内　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）将计算机封装成一个类并实现相关操作。	
		（2）制作一个接口根据传入对象的不同实现不同的操作。	
		（3）利用面向对象的知识封装一个表格操作类。	
		（4）利用面向对象的知识封装一个文件管理系统类。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

14.14　成功可以复制——中国通信设备行业的领跑者任正非

成就：2003 年荣膺网民评选的“2003 年中国 IT 十大上升人物”。2003 年任正非以 11 亿位居福布斯第 273 名。

创始公司：华为技术有限公司

财富巅峰：资产范围 5 亿美元以上

出生日期：1944 年

最高学历：大学毕业

超级语录：世界上一切资源都可能枯竭，只有一种资源可以生生不息，那就是文化。

成功历程

任正非祖籍浙江浦江县，出生在一个贫苦山区的小村庄，任正非的爷爷是一个做火腿的大师傅，任正非父亲的兄弟姊妹都没有读过书。由于爷爷的良心发现，也由于爸爸的执著要求，爸爸才读了书。任母虽然只有高中文化程度，但是受丈夫影响，通过自修，当上了中学教员。虽然是农村，却是一个知识分子家庭。家庭背景是任正非一生第一个决定性因素。中国的知识分子对知识的重视和追求，可谓“贫贱不能移”。在三年自然灾害时期，父母仍然坚持从牙缝里挤出粮食来让孩子读书。任正非凭借其才智和能力，以及对知识的追求，进入了一个科技密集型行业。在文化大革命时，任正非没有放弃学业，还自学了高等数学、逻辑、哲学和三门外语。任正非的知识渊博，见解独到，在他的讲话中体现为旁征博引，一针见血。后来任正非参军了，作为通信兵的他被抽调到一个飞机制造厂，参与一项代号为 011 的军事通讯系统工程。当时中央军委提出要重视高科技的作用。任正非上进好学，有多项发明创造，两次填补了国家空白。因技术方面的多次突破，被选为军方代表，到北京参加全国科学大会。时年 33 岁。



华为产品

任正非从部队转业后选择一家电子公司，而在创办华为时选择高科技含量的电信行业。1992 年，任正非孤注一掷投入 C&C08 机的研发，虽然是形势所逼，也可以看出他对技术的重视。当时身处房地产热和股票热的核心地带，任正非不仅不为所动，而且对于股票和泡沫深恶痛绝。其实在他内心更多的是对知识、技术和真才实学的尊重。

2006 年美国《新闻周刊》说，尽管创立者任正非一直保持低调，华为已经与电讯业的国际几个巨头北方电讯、朗讯科技、阿尔卡特、思科系统站在同一水平线展开竞争，而且华为常常可以从他们之间赢得更多网络运营业务。

如今，华为在全球设立了包括印度、美国、瑞典、俄罗斯以及中国的北京、上海、南京等多个研究所，100000 名员工中的 48%从事研发工作，截至 2008 年 6 月，华为已累计申请专利超过 29666 件，已连续数年

成为中国申请专利最多的单位。华为在全球建立了 100 多个分支机构，营销及服务网络遍及全球，为客户提供快速、优质的服务。



对于人生而言，贫穷是老师，他可以教会人如何去生存。贫穷有压力，可以使你的脊梁比别人更硬，坦然吃苦、不屈不挠。在我国许许多多的民营企业家身上都可以找到贫穷的影子。而他们的吃苦耐劳、坚忍不拔的品质，往往就是创业的精神所在。因此有人将贫穷当作是成功者的“财富”。

第15堂课

PHP与XML技术

（视频讲解：160分钟）

XML语言是目前超于流行的语言，被称为“第二代Web语言”，是Web 2.0中的一项重要技术。无论是RSS订阅、Web Service，还是Ajax无刷新，都和XML有着直接的联系。通过PHP可以对XML进行全面的操作。

学习摘要：

- XML概述
- XML语法
- SimpleXML系列函数操作XML
- RSS阅读器

15.1　XML 的概述

XML（eXtensible Markup Language），扩展性标记语言，它是用来描述其他语言的语言。它允许用户设计自己的标记。XML 是由 W3C（World Wide Web Consortium，互联网联合组织）于 1998 年 2 月发布的一种标准，它的前身是 SGML（Standard Generalized MarkupLanguage，标准通用置标语言）。XML 产生的原因是为了补充 HTML 语言的不足，使网路语言更规范化、多样化。

HTML 语言被称为第一代 Web 语言，现在的版本为 4.0，以后将不再更新，取而代之的是 XHTML。而 XHTML 正是根据 XML 来制定的。XML 的特点介绍如下。

（1）易用性：XML 可以使用多种编辑器来进行编写，包括记事本等所有的纯文本编辑器。

（2）结构性：XML 是具有层次结构的标记语言，包括多层的嵌套。

（3）开放性：XML 语言允许开发人员自定义标记，这使得不同的领域都可以有自己的特色方案。

（4）分离性：XML 语言将数据的显示和数据内容分开保存，各自处理。使得基于 XML 的应用程序可以在 XML 文件中准确高效地搜索相关的数据内容，忽略其他不相关部分。

15.2　XML 语法

XML 语法是 XML 语言的基础，是学好 XML 的前提条件。任何一门语言都有一些共同的东西，同样也有各自的语法特点。下面就来学习 XML 语法特点。

15.2.1　XML 文档结构

在开始讲解 XML 语法之前，先来熟悉一下 XML 的文档结构，代码如下：

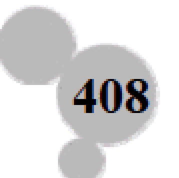
```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<?xml-stylesheet type="text/css" href="Book.css"?>
<!--  下面的标签<计算机图书>就是这个 XML 文档的根目录  -->
<计算机图书>
      <PHP>
            <书名>PHP 程序开发范例宝典</书名>
            <价格  单位="元/本">89.00</价格>
            <出版时间>2007-09-01</出版时间>
      </PHP>
</计算机图书>
```

这就是一个 XML 文档的最基本的元素，包括 XML 声明、处理指令（PI）、注释、元素等，下面就来一一说明。

15.2.2　XML 声明

XML 声明在文档中只能出现一次，而且必须是在第一行，连空行也不允许存在。XML 声明包括了 XML 版本、编码等信息。示例如下：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
```



XML 声明中各部分的含义如表 15.1 所示。

表 15.1　XML 声明中的各部分含义

XML 声明部分	含　　义
<?xml	表示 XML 声明的开始标记。xml 表示该文件是 XML 文件
version="1.0"	XML 的版本说明，是声明中必不可少的属性，而且必须放到第一位
encoding="gb2312"	编码声明。如果声明该属性，那么 XML 默认使用 UTF-8 来解析文档
standalone="yes"	独立声明。如果该属性赋值 yes，那么说明该 XML 文档不依赖于外部文档。如果该属性赋值为 no，则说明该文档有可能依赖于某个外部文档。注意：是有可能的
?>	XML 声明的结束标记

15.2.3　处理指令

“处理指令”，顾名思义，就是处理 XML 文档的指令。有一些 XML 分析器可能对 XML 文档的应用程序不做处理，这时可以指定应用程序按照这个指令信息来处理，然后再传给下一个应用程序。XML 声明其实就是一个特殊的处理指令。处理指令的格式如下：

```
<?处理指令名  处理执行信息?>
```

例如，下面创建一个处理指令，代码如下：

```
<?xml-stylesheet type = "text/css" href="Book.css"?>
```

其中各部分的含义介绍如下。

☑ xml-stylesheet：样式表单处理指令。指明了该 XML 文档所使用的样式表。
☑ type="text/css"：设定了文档所使用的样式是 css。
☑ href="Book.css"：设定了样式文件的地址。

15.2.4　注释

XML 中的注释和 HTML 是一样的，使用“<!--”和“-->”作为开始和结束界定符。注释的用法十分简单，这里只介绍在使用注释时需要注意的几个问题：

☑ 不能出现在 XML 声明之前。
☑ 不能出现在 XML 元素中间，如<computer_book <!--　这是错误的　-->>。
☑ 不能出现在属性列表中。
☑ 不可嵌套注释。
☑ 注释内容可以包含“<”、“>”和“&”这些特殊字符，但不允许有“--”。

15.2.5　XML 元素

元素是每个 XML 文档不可或缺的部分，也是文档内容的基本单元。每个 XML 文档至少要包含一个元素。一般的元素由 3 部分组成，格式如下：

```
<标签>数据内容</标签>
```

其中<标签>为元素的开始标签，</标签>为元素的结束标签，中间的数据内容是元素的值。这里要注意的是标签的写法：

- ☑ <标签>和</标签>都是成对出现的，这是 XML 严格定义的。不允许只有开始标签而没有结束标签，对于空元素，即两个标签之间没有数据，这时可以使用简短形式：<标签/>。
- ☑ 英文标签名称只能以下划线“_”或英文字母开头，中文标签名称只能使用下划线“_”或汉字开头。名称中只能有下划线“_”、连接符“-”、点“.”和冒号“:”几个特殊字符，也可以使用指定字符集下的合法字符。
- ☑ <标签>中不能有空格。<标签>或</ 标签>都是错误的。
- ☑ <标签>对英文大小写很敏感，如<name>和<Name>是两个不同的标签。

15.2.6　XML 属性

XML 属性是 XML 元素中的内容，是可选的。XML 属性和 HTML 中的属性在功能上十分相似，但 XML 属性在格式上更加严格，使用上更加灵活。XML 属性的格式如下：

```
<标签 属性名="属性值" 属性名=""…>内容</标签>
```

这里要注意：

- ☑ 属性名和属性值必须是成对出现的，不像 HTML 中有的属性，可以不需要值而单独存在。对于 XML 来说这是不可能的。如果没有值，写成“属性名=""”也可以。
- ☑ 属性值必须用引号括起来。通常使用双引号，除非属性值本身包含了双引号，这时可以用单引号来代替。

15.2.7　使用 CDATA 标记

在 XML 中对于特殊字符“>”、“<”和“&”的输入需要使用实体引用来处理，实体引用就使用“&…;”的形式来代替那些特殊字符。表 15.2 是 XML 中所用到的实体引用。

表 15.2　XML 中的实体引用

实 体 参 考	字　　符
<	<
>	>
'	'
"	"
&	&

但如果遇到大量的特殊符号需要输入，使用这种方法就不太实际了。XML 中提供了 CDATA（Character data，字符数据）标记。在 CDATA 标记段的内容都会被当作纯文本数据来看。CDATA 标记的格式如下：

```
<![CDATA[
…
]]>
```

例如，创建一个 XML 文件，并且通过实体引用和 CDATA 标记来显示特殊符号。代码如下：

```
<?xml version="1.0" encoding="GB2312"?>
<exam>
    <实体引用>这里必须使用引用“&lt;”、“&gt;”、“&amp;”</实体引用>
    <CDATA 标记>
    <![CDATA[
```



```
		这里可以正常输出“<”、“>”、“&”。
	]]>
	</CDATA 标记>
</exam>
```

注意：在 CDATA 标记段内不允许出现“]]>”，否则，XML 会认为 CDATA 标记段结束。

15.2.8　XML 命名空间

命名空间通过在元素前面增加一个前缀来保证元素和属性的唯一性。它的最重要用途是用于融会不同的 XML 文档。命名空间的格式如下：

```
<标签名称 xmlns:前缀名称="URL">
```

例如，在创建的 XML 文件中，对元素<外语图书>使用命名空间，代码如下：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<外语图书 xmlns:frn="http://www.bccd.com/foreign">
	<frn:English>
		<frn:书名>许国璋英语</frn:书名>
		<frn:价格 货币种类="RMB" 单位="4 本">80.00</frn:价格>
		<frn:出版时间>1996-05-10</frn:出版时间>
	</frn:English>
</外语图书>
```

15.3　SimpleXML 系列函数操作 XML

PHP 对 XML 格式的文档进行操作有很多方法，如使用 XML 语法解析函数、DOMXML 函数或 SimpleXML 函数等。其中，使用 PHP 5 新加入的 SimpleXML 函数操作更简单。本节将详细讲解如何使用 SimpleXML 系列函数对 XML 文档进行读写和浏览。

15.3.1　创建 SimpleXML 对象

使用 SimpleXML 函数首先要创建对象。共有 3 种方法来创建对象，分别介绍如下：

- ☑ Simplexml_load_file()函数，将指定的文件解析到内存中。
- ☑ Simplexml_load_string()函数，将创建的字符串解析到内存中。
- ☑ Simplexml_load_date()函数，将一个使用 dom 函数创建的 domDocument 对象导入到内存中。

例 15.01　使用两个函数分别创建两个对象，并使用 print_r 来输出两个对象，程序代码如下。（**实例位置：光盘\mr\15\sl\15.01**）

```
<?php
header("Content-Type:text/html;charset=utf-8");				//设置编码
/*	第一种方法	*/
$xml_1 = simplexml_load_file("15.01.xml");
print_r($xml_1);
/*	第二种方法	*/
$str = <<<XML
<?xml version='1.0' encoding='utf-8'?>
<Object>
```

```
        <ComputerBook>
                <title>学通 PHP24 堂课</title>
        </ComputerBook>
</Object>
XML;
$xml_2 = simplexml_load_string($str);
echo '<p>';
print_r($xml_2);
?>
```

运行结果为：

SimpleXMLElement Object ([ComputerBook] => SimpleXMLElement Object ([title] =>学通 PHP24 堂课))

SimpleXMLElement Object ([ComputerBook] => SimpleXMLElement Object ([title] =>学通 PHP24 堂课))

可以看到，不同数据源的 XML，只要结构相同，那么输出的结果也是相同的。

注意：第一行中的 header()函数设置了 html 编码。虽然在 XML 文档中设置了编码格式，但那只是针对 XML 文档的，在 html 输出时也要设置编码格式。

15.3.2　遍历所有子元素

创建对象后，就可以使用 SimpleXML 的其他函数来读取数据。使用 SimpleXML 对象中的 children()方法和 foreach 循环语句可以遍历所有子节点元素。

例 15.02　通过 SimpleXML 对象中的 children()函数和 foreach 语句遍历指定 XML 文件中所有子节点元素，程序代码如下。**（实例位置：光盘\mr\15\sl\15.02）**

```
<?php
header('Content-Type:text/html;charset=utf-8');                    //设置编码
/*   创建 XML 格式的字符串   */
$str = <<<XML
<?xml version='1.0' encoding='utf-8'?>
<object>
        <book>
                <computerbook>PHP 从入门到精通</computerbook>
        </book>
        <book>
                <computerbook>学通 PHP24 堂课</computerbook>
        </book>
</object>
XML;
/*   **************************   */
$xml = simplexml_load_string($str);                                 //创建一个 SimpleXML 对象
foreach($xml->children() as $layer_one){                            //循环输出根节点
        print_r($layer_one);                                        //查看节点结构
        echo '<br>';
        foreach($layer_one->children() as $layer_two){              //循环输出第二层根节点
                print_r($layer_two);                                //查看节点结构
                echo '<br>';
        }
}
?>
```

运行结果如图 15.1 所示。

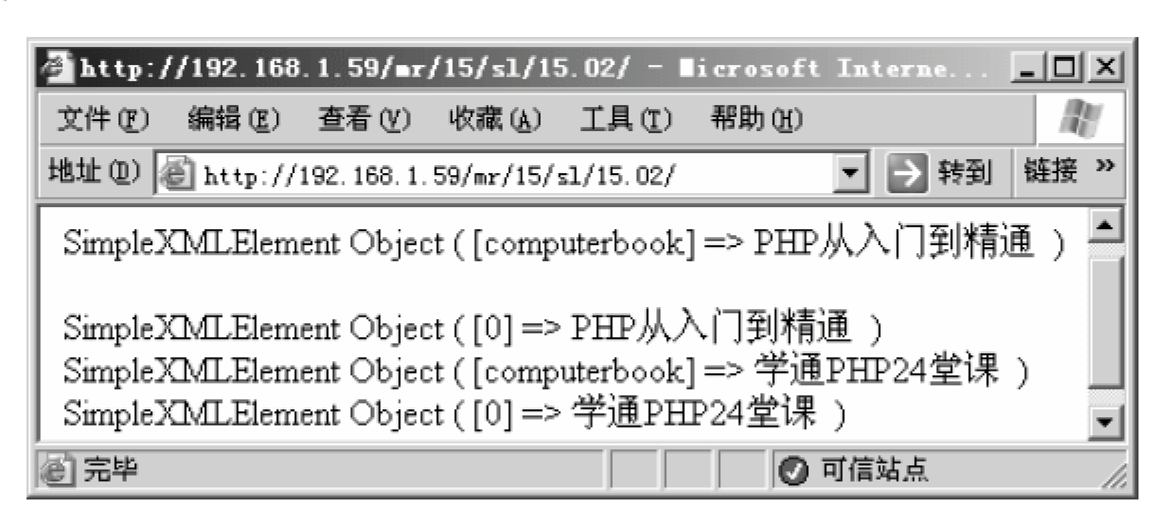


图 15.1　遍历节点

15.3.3　遍历所有属性

使用 SimpleXML 不仅可以遍历子元素，还可以遍历元素中的属性。使用的是 SimpleXML 对象中的 attributes()方法。

例 15.03　通过 SimpleXML 对象中的 attributes 方法来遍历 XML 所有的元素属性，程序代码如下。**（实例位置：光盘\mr\15\sl\15.03）**

```
<?php
header("Content-Type:text/html;charset=utf-8");                    //设置编码
/*   创建 XML 格式的字符串   */
$str = <<<XML
<?xml version='1.0' encoding='utf-8'?>
<object name='commodity'>
	<book type='computerbook'>
		<bookname name='学通 PHP24 堂课'/>
	</book>
	<book type='historybook'>
		<booknanme name='上下五千年'/>
	</book>
</object>
XML;
$xml = simplexml_load_string($str);                                //创建一个 SimpleXML 对象
foreach($xml->children() as $layer_one){                           //循环子节点元素
	foreach($layer_one->attributes() as $name => $vl){             //输出各个节点的属性和值
		echo $name.'::'.$vl;
	}
	echo '<br>';
	foreach($layer_one->children() as $layer_two){                 //输出第二层节点元素
		foreach($layer_two->attributes() as $nm => $vl){           //输出各个节点的属性和值
			echo $nm."::".$vl;
		}
		echo '<br>';
	}
}
?>
```

运行结果如图 15.2 所示。

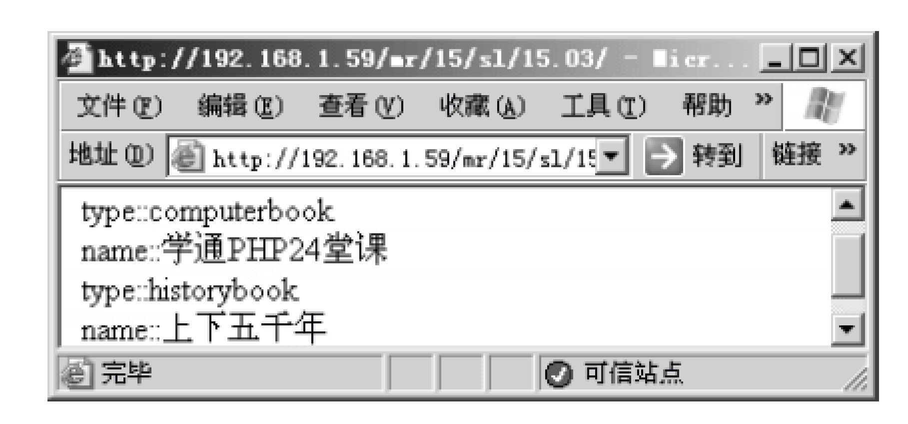


图 15.2　遍历子元素属性

15.3.4　访问特定节点元素和属性

通过 SimpleXML 对象除了可以使用上面两种方法来遍历所有的节点元素和属性，还可以访问特定的数据元素。SimpleXML 对象可以通过子元素的名称对该子元素赋值，或使用子元素的名称数组来对该子元素的属性赋值。

例 15.04　通过 SimpleXML 对象直接对 XML 元素和属性进行访问，程序代码如下。（**实例位置：光盘\mr\15\sl\15.04**）

```
<?php
header('Content-Type:text/html;charset=utf-8');                    //设置编码
/*   创建 XML 格式的字符串   */
$str = <<<XML
<?xml version='1.0' encoding='utf-8'?>
<object name='商品'>
    <book>
        <computerbook>学通 PHP24 堂课</computerbook>
    </book>
    <book>
        <computerbook name='PHP 项目开发全程实录'/>
    </book>
</object>
XML;
/*   **************************   */
$xml = simplexml_load_string($str);                                //创建 SimpleXML 对象
echo $xml['name'].'<br>';                                          //输出根元素的属性 name
echo $xml->book[0]->computerbook.'<br>';                           //输出子元素中 computerbook 的值
echo $xml->book[1]->computerbook['name'].'<br>';                   //输出 computerbook 的属性值
?>
```

运行结果如图 15.3 所示。

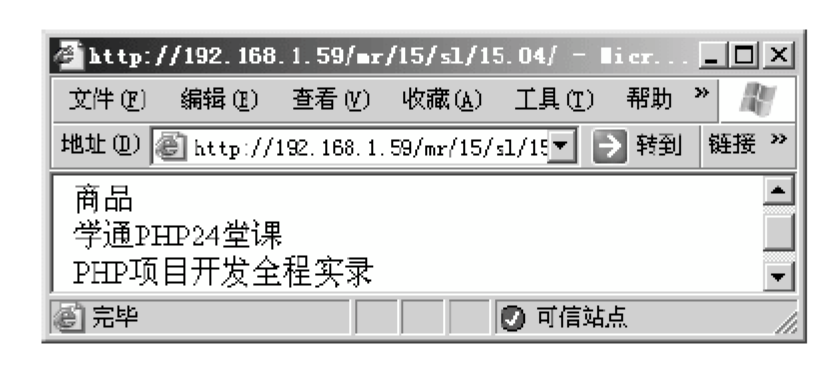


图 15.3　访问特定的节点元素和属性

15.3.5　修改 XML 数据

对 XML 数据的修改，同读取 XML 数据相似。如例 15.04，在访问特定元素或属性时，也可以对其进行



修改操作。

例 15.05　首先读取 XML 文档，然后输出根元素的属性 name，接着修改子元素 computerbook，最后输出修改后的值，程序代码如下。**（实例位置：光盘\mr\15\sl\15.05）**

```
<?php
header('Content-Type:text/html;charset=utf-8'); /*   设置编码格式   */
/*   创建 XML 格式的字符串   */
$str = <<<XML
<?xml version='1.0' encoding='utf-8'?>
<object name='商品'>
     <book>
          <computerbook type='PHP 入门应用'>学通 PHP24 堂课</computerbook>
</book>
</object>
XML;
/*   **************************   */
/*   创建 SimpleXML 对象   */
$xml = simplexml_load_string($str);
/*   输出根目录属性 name 的值   */
echo $xml['name'].'<br />';
/*   修改子元素 computerbook 的属性值 type   */
$xml->book->computerbook['type'] = 'PHP 程序员必备工具';
/*   修改子元素 computerbook 的值   */
$xml->book->computerbook = 'PHP 函数参考大全';
/*   输出修改后的属性和元素值   */
echo $xml->book->computerbook['type'].' => ';
echo $xml->book->computerbook;
?>
```

运行结果如图 15.4 所示。

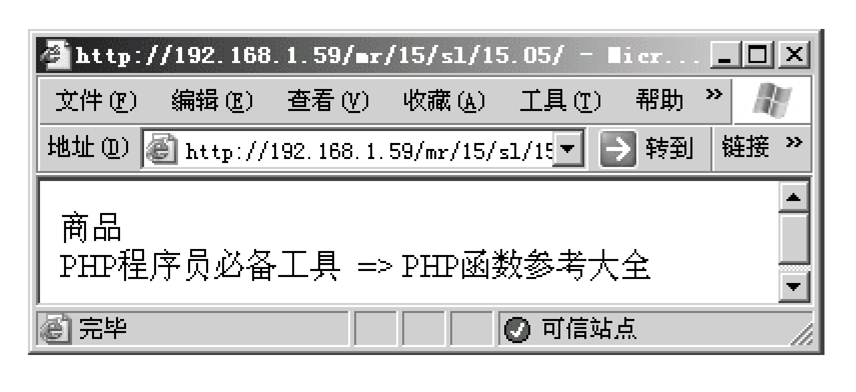


图 15.4　修改元素和属性值

15.3.6　保存 XML 文档

数据在 SimpleXML 对象中所做的修改，其实是在系统内存中做的改动，而原文档根本没有变化。当关掉网页，或清空内存时，数据又会恢复。要保存一个修改过的 SimpleXML 对象，可以使用 asXML()方法来实现。该方法可以将 SimpleXML 对象中的数据格式化为 XML 格式，然后再使用 file 函数中的写入函数，将数据保存到 XML 文件中。

例 15.06　首先从 15.06.xml 文档中生成 SimpleXML 对象，然后对 SimpleXML 对象中的元素进行修改，最后将修改后的 SimpleXML 对象再保存到 15.06.xml 文档中，程序代码如下。**（实例位置：光盘\mr\15\sl\15.06）**

15.06.xml 文档的代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<object name="商品">
```

```
    <book>
        <computerbook type="PHP 入门">学通 PHP24 堂课</computerbook>
    </book>
</object>
```

index.php 文件的代码如下：

```
<?php
header('Content-Type:text/html;charset=utf-8'); /*  设置编码格式  */
/*  创建 SimpleXML 对象  */
$xml = simplexml_load_file('15.06.xml');
/*  修改 XML 文档内容  */
$xml->book->computerbook['type'] = 'PHP 程序员必备工具';
$xml->book->computerbook = 'PHP 函数参考大全';
/*  格式化对象$xml  */
$modi = $xml->asXML();
/*  将对象保存到 10.xml 文档中  */
file_put_contents('15.06.xml',$modi);
/*  重新读取 xml 文档  */
$str = file_get_contents('15.06.xml');
/*  输出修改后的文档内容  */
echo $str;
?>
```

运行结果如图 15.5 所示。

```
<?xml version="1.0" encoding="utf-8" ?>
- <object name="商品">
 - <book>
     <computerbook type="PHP程序员必备工具">PHP函数参考大全</computerbook>
   </book>
 </object>
```

图 15.5　保存 SimpleXML 对象

15.4　RSS 阅读器

RSS 为 Really Simple Syndication（简易供稿）的缩写，是某一站点用来和其他站点之间共享内容的一种简易方式，也叫聚合内容。网络用户可以在客户端借助于支持 RSS 的新闻聚合工具软件（如 SharpReader NewzCrawler、FeedDemon RSSReader），在不打开网站内容页面的情况下阅读支持 RSS 输出的网站内容。

从 RSS 阅读者的角度来看，完全不必考虑它到底是什么意思，只要简单地理解为一种方便的信息获取工具即可。RSS 获取信息的模式与加入邮件列表（如电子杂志和新闻邮件）获取信息有一定的相似之处，也就是可以不必登录各个提供信息的网站而通过客户端浏览方式（称为“RSS 阅读器”）或者在线 RSS 阅读方式浏览内容。

15.4.1　RSS 概述

RSS 最早的版本是 0.90，是由 Netscape 公司设计的，目的是用来建立一个整合了各主要新闻站点内容的门户，但是 0.90 版本的 RSS 规范过于复杂，而一个简化的 RSS 0.91 版本也随着 Netscape 公司对该项目的放弃而于 2000 年暂停。



不久，一家专门从事博客写作软件开发的公司 UserLand 接手了 RSS 0.91 版本的发展，并把它作为其博客写作软件的基础功能之一继续开发，逐步推出了 0.92、0.93 和 0.94 版本。随着网络博客的流行，RSS 作为一种基本的功能也被越来越多的网站和博客软件支持。

在 UserLand 公司接手并不断开发 RSS 的同时，很多的专业人士认识到需要通过一个第三方、非商业的组织，把 RSS 发展成为一个通用的规范，并进一步标准化。于是 2001 年一个联合小组在 0.90 版本 RSS 的开发原则下，以 W3C 新一代的语义网技术 RDF（Resource Description Framework）为基础，对 RSS 进行了重新定义，发布 RSS 1.0，并将 RSS 定义为“RDF Site Summary”。

但是这项工作没有与 UserLand 公司进行有效的沟通，UserLand 公司也不承认 RSS 1.0 的有效性，并坚持按照自己的设想进一步开发出 RSS 的后续版本，到 2002 年 9 月发布了最新版本 RSS 2.0，UserLand 公司将 RSS 定义为“Really Simple Syndication”。

到目前为止，RSS 从 0.91 起到 2.01，一共有 7 个版本之多。各个版本除 1.0 以外大同小异。

15.4.2　RSS 2.0 的标准

在 RSS 阅读器中，读取的是站点中一个标准格式的 RSS 2.0 文件。所以让站点支持 RSS 阅读，标准 RSS 2.0 格式文件的创建是关键，这里先不讲解如何来创建这个 RSS 2.0 文件，而是先来了解一下 RSS 2.0 文件的结构，分析一下标准的 RSS 2.0 格式文件都包括哪些元素。

RSS 2.0 结构可以划分为 3 个部分。

- ☑　RSS 的根元素 rss。
- ☑　在根元素下的 channel 元素（表示新闻频道的特性）。
- ☑　在根元素下 channel 元素之后出现的 item 元素（表示新闻条目）。

1．rss 的根元素

rss 元素是 RSS 的根元素，必须具有 version 属性，可以包含任何名称空间的其他属性。在该元素下的子元素有两个，即 channel（频道，只能出现一次）和 item（条目，可以出现多次）。

2．channel（频道）元素及其子元素

channel 元素表示网站上所发布的新闻栏目，该元素的子元素如表 15.3 所示。

表 15.3　channel 元素的可用子元素

元 素 名 称	说　明
title	网站上新闻频道的标题，该标题应该与 html 网站的标题保持一致
link	可以到达此频道相关的 HTML 网页的超链接
description	频道的描述
language	频道对应的语种
copyright	网站著作权的声明
managingEditor	频道内容编辑的电子邮件地址
webMaster	维护网站频道的主要技术人员
pubDate	发布 RSS 的时间
lastBuilDate	最后更新网站内容的时间
category	该栏目所属的类别，此元素包含 domain 属性

续表

元素名称	说明
generator	生成 RSS 栏目的程序
docs	与 RSS 所执行的版本标准对应的 URL（很少用）
cloud	表示应用程序注册订阅此频道的信息，RSS 应用程序使用此元素的信息，自动检查 RSS 更新。此元素的子元素介绍如下。 ☑ domain：自动下载 RSS 时的域名。 ☑ port：下载 RSS 时所采用的端口。 ☑ path：RSS 在域名下的路径。 ☑ registerProcedure：订阅 RSS 时所执行的程序。 ☑ protocol：传输下载 RSS 时所用的网络协议，如 HTTP 等。
ttl	RSS 内容在客户浏览器中缓存的时间
image	与频道对应的图标
textInput	可以用来接受文本的输入框，其子元素介绍如下。 ☑ title：提交按钮上的文字。 ☑ description：对此文本框的描述。 ☑ name：文本框的名称。 ☑ link：服务器上处理文本请求的应用程序路径。
skipHours	RSS 下载程序不检查 RSS 更新的时间，此元素包含 hour 子元素（最多 24 个），表示不检查 RSS 更新的钟点，hour 元素值所代表的每个时间对应格林威治时间（Greenwich Mean Time，GMT）的小时
skipDays	RSS 下载程序可以跳跃不检查 RSS 更新的时期，此元素包含 day 子元素（最多 7 个），表示不检查 RSS 更新的日期，day 元素的值用 Monday、Tuesday、Wednesday、Thursday、Friday、Saturday 和 Sunday 分别代表星期一到星期日

3．item（项目）元素及其子元素

item 元素表示网站上所发布的新闻，该元素的子元素如表 15.4 所示。

表 15.4　item 元素的可用子元素

元素名称	说明
title	新闻标题
link	阅读新闻正文的超链接
description	新闻内容的摘要
author	本条目作者的电子邮件地址
category	此新闻所属的类别，包含 domain 属性（关于此类别的领域名称）
comments	此新闻条目对应的评论网页的超链接
enclosure	与此条目相关的多媒体的内容，此元素包含 url（下载附件的超文本链接路径）、length（附件字节大小）和 type（附件的 MIME 媒体类型）属性
guid	全球唯一标识符（Globally Unique Identifier）或永久的 URL 值，此元素包含 isPermaLink（是否为永久的 URL 值）属性
pubDate	发布此新闻的时间
source	此新闻的来源，此元素包含 url（此新闻所引用的频道的路径）属性

上述 3 个小节中的内容就是 RSS 2.0 中包含的元素，其中在生成 RSS 文件时，不会应用到所有的元素，而是根据实际的需求应用不同的元素。

而在 RSS 阅读器中读取的 RSS 文件中，包含的元素介绍如下：

- ☑ rss 根元素。
- ☑ channel（频道）元素。channel 元素包含 3 个子元素，分别介绍如下。
  - ➢ title：网站新闻频道的标题。
  - ➢ link：此频道相关的 HTML 超链接。
  - ➢ description：频道内容的摘要。
- ☑ item（项目）元素。item 元素包含 4 个子元素，分别介绍如下。
  - ➢ title：新闻标签。
  - ➢ link：阅读新闻正文的超链接。
  - ➢ description：新闻内容的摘要。
  - ➢ pubDate：新闻的发布日期。

注意：这里采用 RSS 2.0 作为 RSS 文件的标准格式。

你问我答：RSS 的发展趋势。

随着越来越多的站点对 RSS 的支持，RSS 已经成为目前最成功的 XML 应用。RSS 搭建了信息迅速传播的一个技术平台，使得每个人都成为潜在的信息提供者。相信用户很快就会看到大量基于 RSS 的专业门户、聚合站点和更精确的搜索引擎。

15.4.3　创建支持 RSS 阅读的站点

在了解了 RSS 2.0 的标准格式之后，下面创建一个支持 RSS 阅读功能的站点。因为 PHP 是面向 Web 开发，而 RSS 阅读器这个模块似乎更适合通过应用程序来完成。所以作为 PHP 程序员更应该了解的是如何让网站支持 RSS 阅读，而不是如何来更好地去创建开发 RSS 阅读器。还有一个更直接的原因，如果想使用 RSS 阅读器，从网上下载来得更直接。

例 15.07　下面编写一个支持 RSS 阅读的站点，其开发步骤如下。**（实例位置：光盘\mr\15\sl\15.07）**

1．创建支持 RSS 阅读的站点

要让站点支持 RSS 阅读，必须考虑以下几个问题：

（1）将什么样的数据作为 RSS 阅读的内容。

（2）RSS 阅读器读取的是什么格式的文件。

（3）如何将数据库中的数据写入到文件中。

（4）如何才能保证 RSS 阅读器中读取的是最新的数据。

带着上述的疑问创建一个能够支持 RSS 阅读的程序，这里创建的是一个用于统计编程技巧的程序，通过该程序将每个人在编程中总结出的经验技巧进行统计，存储到数据库中，然后再输出这些技巧让大家共享资源。运行结果如图 15.6 所示。

有关该程序的创建这里将不作讲解，完整代码请参考本书附带光盘中的内容。这里将编程技巧的标题和内容作为 RSS 阅读的内容，所以接下来要做的就是如何将编程技巧的标题和内容添加到 RSS 文件中，并且保证 RSS 文件中的数据是最新的。



图 15.6　编程技巧整合的运行结果

2．动态创建 RSS 文件

首先，保证 RSS 文件中的内容是最新的，那么就要在数据被添加后，直接更新 RSS 文件，只有这样才能保证 RSS 文件中的数据也是最新的。

在 index_ok.php 中执行向数据库中添加编程技巧的操作，并调用 rss.php 文件重新生成一个标准格式的 XML 文件，更新 XML 文件中的数据。index _ok.php 的关键代码如下：

```
<?php
header("Content-Type:text/html; charset=utf-8");                    //设置编码格式
include_once("conn/conn.php");                                      //包含数据库连接文件
$dates=date("Y-m-d");                                               //设置时间
if($_POST['submit']=="提交"){                                       //判断是否执行提交操作
    $query="insert into tb_rss_database (tb_rss_subject,tb_rss_content,tb_rss_author,tb_rss_ department,tb_
rss_date)values('".$_POST['zhuti']."','".$_POST['neirong']."','".$_POST['user']."','".$_POST['bumen']."','$dates')";
    //定义添加语句
    $result=mysql_query($query,$conn);                              //执行添加操作
    if($result){
        include_once("rss.php");                                    //包含生成 RSS 的文件，重新生成 RSS 文件
        echo "<script> alert ('文件上传成功!'); window.location.href='index.php';</script>";
    }else{
        echo "<script> alert ('文件上传失败!'); window.location.href='index.php';</script>";
    }
}
?>
```

然后，将数据添加到 RSS 文件中，并且生成一个标准格式的 RSS 2.0 文件，文件的后缀是.xml。创建 XML 文件的操作是在 rss.php 文件中进行的，其操作步骤如下：

（1）通过服务器变量获取头信息的内容和当前页面前一页的 URL 地址，以及当前的时间。代码如下：

```
$server_ip=$_SERVER['SERVER_NAME'];                                 //获取服务器 IP 地址
$request_url=str_ireplace("index_ok.php","look_content.php",$_SERVER['REQUEST_URI']);
                                                                    //获取实例的相对路径
$http="http://".$server_ip.$request_url;                            //定义一个访问路径
$date_time=date("Y-m-d H:i:s");
```

（2）通过 new DomDocument 创建 DOM 对象，创建 XML 文件的根节点，并且为根节点添加属性及属性值。代码如下：



```
$dom = new DomDocument('1.0','utf-8');                                //创建 DOM 对象
$object = $dom->createElement('rss');                                 //创建根节点 rss
$dom->appendChild($object);                                           //将创建的根节点添加到 dom 对象中
    $type1 = $dom->createAttribute('xmlns:rdf');                      //创建一个节点属性 xmlns:rdf
    $object->appendChild($type1);                                     //将属性追加到 rss 根节点中
        $type1_value = $dom->createTextNode('http://www.w3.org/1999/02/22-rdf-syntax-ns#');
                                                                      //创建一个属性值
        $type1->appendChild($type1_value);                            //将属性值赋给属性 xmlns:rdf
    $type2 = $dom->createAttribute('xmlns:dc');                       //创建一个节点属性 xmlns:dc
    $object->appendChild($type2);                                     //将属性追加到 rss 根节点中
        $type2_value = $dom->createTextNode('http://purl.org/dc/elements/1.1/'); //创建一个属性值
        $type2->appendChild($type2_value);                            //将属性值赋给属性 xmlns:dc
    $type3 = $dom->createAttribute('xmlns:taxo');                     //创建一个节点属性 xmlns:taxo
    $object->appendChild($type3);                                     //将属性追加到 rss 根节点中
        $type3_value = $dom->createTextNode('http://purl.org/rss/1.0/modules/taxonomy/'); //创建一个属性值
        $type3->appendChild($type3_value);                            //将属性值赋给属性 xmlns:taxo
    $type4 = $dom->createAttribute('version');                        //创建一个节点属性 version
    $object->appendChild($type4);                                     //将属性追加到 rss 根节点中
        $type4_value = $dom->createTextNode('2.0');                   //创建一个属性值
        $type4->appendChild($type4_value);                            //将属性值赋给属性 version
```

（3）创建 channel 节点，这是 RSS 2.0 中的 channel 元素及其子元素。其中包括 4 个子元素，即 title、link、description 和 dc:creator。代码如下：

```
$channel = $dom->createElement('channel');                            //创建节点 channel
    $object->appendChild($channel);                                   //将节点 channel 追加到根节点 rss 下
        $title = $dom->createElement('title');                        //创建节点 title
        $channel->appendChild($title);                                //将节点追加到 channel 节点下
            $title_value = $dom->createTextNode('明日科技');          //创建元素值
            $title->appendChild($title_value);                        //将值赋给 title 节点
        $link = $dom->createElement('link');                          //创建节点 link
        $channel->appendChild($link);                                 //将节点追加到 channel 节点下
            $link_value = $dom->createTextNode('http://www.mingrisoft.com');//创建元素值
            $link->appendChild($link_value);                          //将值赋给 link 节点
        $description = $dom->createElement('description');            //创建节点 description
        $channel->appendChild($description);                          //将节点追加到 channel 节点下
            $description_value = $dom->createTextNode('明日科技'); //创建元素值
            $description->appendChild($description_value);            //将值赋给 description 节点
        $dc_creator = $dom->createElement('dc:creator');              //创建节点 dc:creator
        $channel->appendChild($dc_creator);                           //将节点追加到 channel 节点中
            $dc_creator_value = $dom->createTextNode('http://www.mingrisoft.com');//创建元素值
            $dc_creator->appendChild($dc_creator_value);              //将值赋给 dc:creator 节点
```

（4）创建 item 节点。从数据库中读取出编程技巧的数据，并且应用 while 循环输出数据库中的数据，将其作为 item 节点中子节点的元素值。这里创建的是 RSS 2.0 中的 item 元素及其子元素，其中包括 4 个子元素，即 title、link、description 和 pubDate。代码如下：

```
$sql=mysql_query("select * from tb_rss_database order by tb_rss_id desc");   //从数据库中读取数据
while($myrow=mysql_fetch_array($sql)){                                //循环输出数据库中数据
        $item = $dom->createElement('item');                          //创建节点 item
        $channel->appendChild($item);                                 //将 item 追加到 channel 节点下
            $item_title = $dom->createElement('title');               //创建 title 节点
            $item->appendChild($item_title);                          //将节点追加到 item 节点下
```

```
        $item_link = $dom->createElement('link');                    //创建 link 节点
        $item->appendChild($item_link);                              //将节点追加到 item 节点下
        $item_description = $dom->createElement('description');      //创建 description 节点
        $item->appendChild($item_description);                       //将节点追加到 item 节点中
        $item_pubDate = $dom->createElement('pubDate');               //创建节点 pubDate
        $item->appendChild($item_pubDate);                           //将节点追加到 item 节点下
            $title_value = $dom->createTextNode($myrow['tb_rss_subject']);            //创建元素值
            $item_title->appendChild($title_value);                  //将值赋给 title 节点
            $link_value = $dom->createTextNode($http."?lmbs=".$myrow['tb_rss_id']);   //创建元素值
            $item_link->appendChild($link_value);                    //将值赋给 link 节点
            $description=mb_substr($myrow['tb_rss_content'],0,80,"utf-8");   //截取该字段中的前 80 个字符
            $description_value = $dom->createTextNode($description);        //创建元素值
            $item_description->appendChild($description_value);             //将值赋给 description 节点
            $pubDate_value = $dom->createTextNode($date_time);              //创建元素值
            $item_pubDate->appendChild($pubDate_value);                     //将值赋给 pubDate 节点
}
```

（5）通过$dom->saveXML()方法生成 XML 文档，并应用 file_put_contents()函数将生成的 XML 文档保存到 Rss.xml 文件中，代码如下：

```
$modi = $dom->saveXML();                    //生成 xml 文档
file_put_contents('Rss.xml',$modi);         /*  将对象保存到 Rss.xml 文档中  */
```

至此，一个支持 RSS 阅读的站点创建完毕，只要在站点中添加新的数据，存储在根目录下的 xml 文件就会被更新一次，从而保证了 RSS 阅读器中能读取到站点中最新的数据。

15.5　照猫画虎——基本功训练

15.5.1　基本功训练 1——动态创建 XML 文档

视频讲解：光盘\mr\15\lx\动态创建 XML 文档.exe

实例位置：光盘\mr\15\zmhh\01

使用 SimpleXML 对象可以十分方便地读取和修改 XML 文档，但却无法动态建立 XML。这时就需要使用 DOM 来实现。DOM（Document Object Model，文档对象模型），通过树形结构模式来遍历 XML 文档。使用 DOM 遍历文档的好处是不需要标记就可以显示全部内容。但缺点同样明显，就是非常消耗内存。

通过 DOM 类库创建 XML 文档应用到的方法如下：

（1）通过 new DomDocument()来实例化一个 DomDocument 对象。语法如下：

```
$dom = new DomDocument('1.0','gb2312');                      //创建 DOM 对象
```

（2）通过$dom->createElement 创建一个新元素节点。语法如下：

```
class DomDocument {
DomElement createElement ( string name [, string value] )
}
```

参数 name 是创建的元素的名称，参数 value 是创建的元素的值。成功则返回一个元素节点对象，否则返回一个错误。

（3）通过$dom->appendChild 实现元素节点的附加。语法如下：

```
class DomNode {
```



```
DomNode appendChild ( DOMNode newnode )
}
```

参数 newnode 是将要附加的节点，返回更多的节点。

（4）通过$dom->createAttribute 创建一个新的属性。语法如下：

```
class DomDocument {
DomAttr createAttribute ( string name )
}
```

参数 name 是属性的名称，成功则返回一个属性对象，否则返回一个错误。

（5）通过$dom->createTextNode 创建一个属性值。语法如下：

```
class DomDocument {
DomText createTextNode ( string content )
}
```

参数 content 是属性的值，成功则返回一个属性值对象，否则返回一个错误。

（6）通过$dom->saveXML 生成一个 XML 文档。语法如下：

```
class DomDocument {
string saveXML ( [DOMNode node [, int options]] )
}
```

参数 node 指定生成的节点变量名称，参数 options 是附加的值。成功则返回一个 XML 文档，否则返回一个错误。

上述 6 种方法是在创建 XML 文档中需要使用的。下面应用上述的方法创建一个 XML 文档，运行结果如图 15.7 所示。

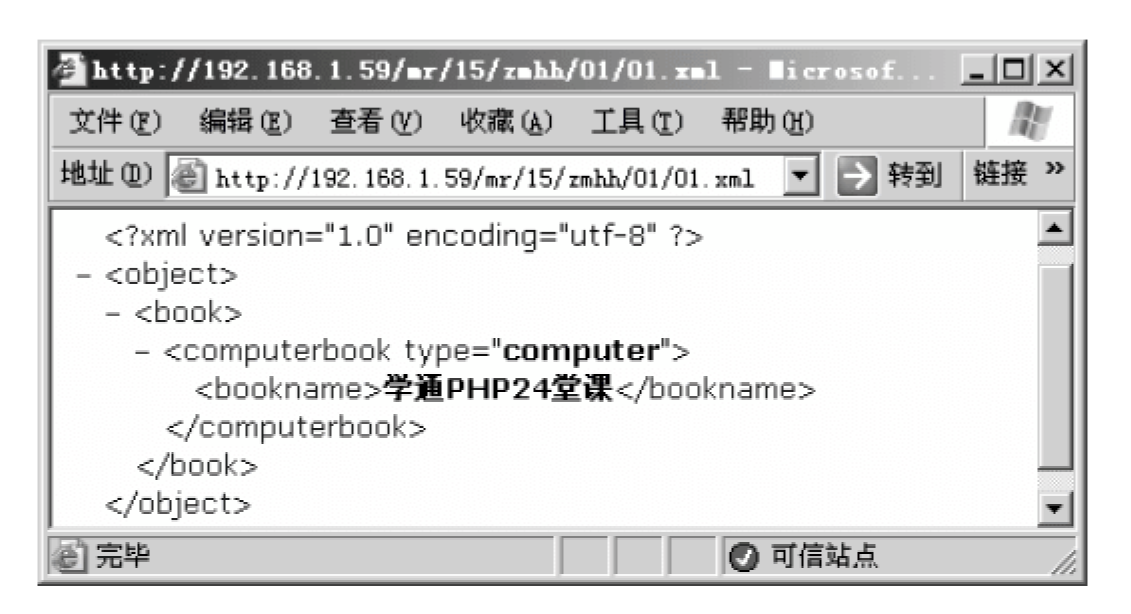


图 15.7　动态创建 XML 文档

关键代码如下：

```
<?php
header('Content-Type:text/html;charset=utf-8');             //设置编码格式
$dom = new DomDocument('1.0','utf-8');                      //创建 DOM 对象
$object = $dom->createElement('object');                    //创建根节点 object
$dom->appendChild($object);                                 //将创建的根节点添加到 dom 对象中
$book = $dom->createElement('book');                        //创建节点 book
$object->appendChild($book);                                //将节点 book 追加到 dom 对象中
$computerbook = $dom->createElement('computerbook');        //创建节点 computerbook
$book->appendChild($computerbook);                          //将 computerbook 追加到 dom 对象中
$type = $dom->createAttribute('type');                      //创建一个节点属性 type
$computerbook->appendChild($type);                          //将属性追加到 computerbook 元素后
$type_value = $dom->createTextNode('computer');             //创建一个属性值
$type->appendChild($type_value);                            //将属性值赋给 type
$bookname = $dom->createElement('bookname');                     //创建节点 bookname
```

```
$computerbook->appendChild($bookname);                 //将节点追加到 dom 对象中
$bookname_value = $dom->createTextNode('学通 PHP24 堂课');   //创建元素值
$bookname->appendChild($bookname_value);               //将值赋给节点 bookname
$xml_file=$dom->saveXML();
echo $xml_file;                                        //输出 XML 文件
file_put_contents('01.xml',$xml_file);                 //将 XML 文档保存到指定的文件中
?>
```

照猫画虎：用上述介绍的方法，将从数据库中读取出的数据生成一个标准的 RSS 2.0 格式的 XML 文档。（20 分）（实例位置：光盘\mr\15\zmhh\01_zmhh）

15.5.2　基本功训练 2——动态读取 XML 文档

视频讲解：光盘\mr\15\lx\动态读取 XML 文档.exe

实例位置：光盘\mr\15\zmhh\02

既然通过 XML 可以将数据存储在一个文档中，那么就一定有方法可以读取该文档中的数据。本实例将介绍在 PHP 中如何读取 XML 文档，其主要应用 DOM 类库中的 getElementsByTagName()方法。运行本实例，通过 PHP 动态读取 XML 文档中的数据，运行结果如图 15.8 所示。



图 15.8　PHP 读取 XML 文件

本实例主要应用 DOM 类库中的 getElementsByTagName()方法读取 XML 文档，并以 for 和 foreach 语句作为辅助，循环输出 XML 文档中的数据。

DOMDocument->getElementsByTagName()方法用来获取标签的名称，返回值为一个新的 DOMNodeList 对象。语法如下：

```
class DOMDocument {
     DOMNodeList getElementsByTagName ( string name )
}
```

参数 name 是 XML 文档中节点的名称。

本实例的所有功能都是在 index.php 文件中实现的，具体步骤如下：



（1）编写 PHP 脚本语句，创建一个 Read_XML()类，继承 DOMDocument 类库。在类文件中定义了两个函数：一个是构造函数，通过 load()方法读取存储在根目录下的 Rss.xml 文件；另一个是 show_xml()函数，应用 getElementsByTagName()方法读取 XML 文档中数据，并通过 for 和 foreach 语句循环输出数据。关键代码如下：

```
<?php
class Read_XML extends DOMDocument{                                    //创建一个类
private $root;
public function __construct(){                                         //构造函数
    parent ::__construct();
     $this->load("Rss.xml");                                           //读取指定的 XML 文件
}
public function show_xml(){                                            //输出 XML 文件中的内容
     $root=$this->documentElement;                                     //定义根目录
      $node_record=$this->getElementsByTagName("item");                //获取节点 item 下的标签
      $node_record_length=$node_record->length;                        //获取标签的数量
      for($i=0;$i<$node_record->length;$i++){                          //根据标签的数量执行 for 循环
          $k=0;
          foreach($node_record->item($i)->childNodes as $data){        //通过 foreach 语句输出标签中的数据
              $field[$k]= $data->textContent;
                $k++;
          }
          echo '<table width="672" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor=
"#999999"><tr>';
          echo '<td width="102" height="22" align="right" bgcolor="#FFFFFF">标题：</td>';
          echo '<td width="368" bgcolor="#FFFFFF"> '.$field[0].'</td>';
          echo '<td width="184" align="center" bgcolor="#FFFFFF">时间：'.$field[3].'</td>';
          echo '</tr><tr>';
          echo '<td colspan="3" height="40"; style="padding-left:5px; padding-right:5px; padding-top:5px;
padding-bottom:5px;" bgcolor="#FFFFFF">'.$field[2].'</td>';
          echo '</tr><tr>';
        echo '<td height="25" align="right" bgcolor="#FFFFFF">URL:</td>';
          echo '<td colspan="2" bgcolor="#FFFFFF">  '.$field[1].'</td>';
          echo '</tr></table>';
        echo '</table>';
      }
}
}
?>
```

（2）创建 table 表格，添加查看 XML 文件的超链接。实例化 Read_XML 类，并执行 show_xml()方法，完成 XML 文件的读取。关键代码如下：

```
<table width="672" height="30" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor=
"#999999">
  <tr>
    <td height="30" align="center" bgcolor="#FFFFFF">PHP 读取 XML 文件     
   <a href="Rss.xml" class="STYLE3">查看 XML 文件内容</a></td>
  </tr>
  <?php
  $Read_XMl=new Read_XML;                  //类的实例化
  $Read_XMl->show_xml();                   //执行类中方法
```

```
  ?>
</table>
```

（3）在实例的根目录中创建一个 Rss.xml 文件，作为读取的内容。这里使用的是 01_diy 中创建的 XML 文件。

照猫画虎：通过 SimpleXML 函数动态读取 XML 文档。（20 分）（实例位置：光盘\mr\15\zmhh\02_zmhh）

15.5.3　基本功训练 3——通过 XPath 查询 XML 中的数据

视频讲解：光盘\mr\15\lx\通过 XPath 查询 XML 中的数据.exe

实例位置：光盘\mr\15\zmhh\03

在 PHP 中实现对 XML 的查询操作，可以快速地从 XML 中获取指定的内容。运行本实例，通过 DOM 类库中的 DOMXPath->query()方法完成对 XML 的查询操作，运行结果如图 15.9 所示。



图 15.9　PHP 查询 XML 文件

应用 DOM 类库中的 DOMXPath->query()方法完成对 XML 文件的查询操作，然后通过 for 和 foreach 语句循环输出查询结果。

（1）DOMXPath->query()方法用来查询出 XML 文件中符合条件的特定节点，返回值为一个节点列表对象。语法如下：

```
class DOMXPath {
     DOMNodeList query ( string expression [, DOMNode contextnode] )
}
```

其参数 expression 是执行的查询语句。成功返回一个节点列表对象 DOMNodeList，失败返回一个空的节点列表对象 DOMNodeList。

本实例中使用的查询语句如下：

```
"//item[title='$search_title']"
```

查询出 item 节点中，title 子节点的值等于变量$search_title 的特定节点。

（2）$node_Record 对象用来查询返回的节点列表对象，通过 for 和 foreach 语句循环输出$node_Record 对象中的数据。

DOMDocument->getElementsByTagName()方法用来获取标签的名称，返回值为一个新的 DOMNodeList 对象。语法如下：

```
class DOMDocument {
     DOMNodeList getElementsByTagName ( string name )
}
```

参数 name 是 XML 文档中节点的名称。

通过 XPath 查询 XML 中的数据的操作步骤如下：

（1）在本实例中，首先通过 JQuery 技术动态输出整个 XML 文档中数据。应用存储在 js 文件夹下的 fun.



js 脚本文件和 jquery-1.3.2.js 框架包。

（2）创建 index.php 文件。首先，创建 Search_XML 类，继承 DOM 类库，定义构造方法读取指定的 XML 文件；定义 Select_XML()方法，通过 DOMXPath->query()完成对 XML 的查询操作。关键代码如下：

```
<?php
class Search_XML extends DomDocument{                                   //创建类，继承 DOM 类库
    public function __construct() {                                     //构造方法，打开指定的 XML 文件
        parent:: __construct();
        $this->load("Rss.xml");
    }
    public function Select_XML($search_title){                  //定义方法，执行查询操作，参数查询的关键字
        $xpath = new DOMXPath($this);                                   //应用 DOMXPath 方法进行查询
        $node_Record = $xpath->query("//item[title='$search_title']");  //执行查询
        for($i=0;$i<$node_Record->length;$i++){                         //循环输出查询结果
            $g=0;
            foreach($node_Record->item($i)->childNodes as $node){ //读取数组中的数据
                $field[$g]=$node->textContent;
                $g++;
            }
            echo '<table width="672" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF"
bgcolor="#999999"><tr>';
            echo '<td width="102" height="22" align="right" bgcolor="#FFFFFF">标题：</td>';
            echo '<td width="368" bgcolor="#FFFFFF"> '.$field[0].'</td>';
            echo '<td width="184" align="center" bgcolor="#FFFFFF">时间：'.$field[3].'</td>';
            echo '</tr><tr>';
            echo '<td colspan="3" height="40" style="padding-left:5px; padding-right:5px; padding-top:5px;
padding-bottom:5px;" bgcolor="#FFFFFF">  '.$field[2].'</td>';
            echo '</tr><tr>';
        echo '<td height="25" align="right" bgcolor="#FFFFFF">URL:</td>';
            echo '<td colspan="2" bgcolor="#FFFFFF">  '.$field[1].'</td>';
            echo '</tr></table>';
        echo '</table>';
        }
    }
}
?>
```

然后，通过 Script 脚本标记调用 fun.js 和 jquery-1.3.2.js 脚本文件，完成 XML 文件的读取。关键代码如下：

```
<script src="js/jquery-1.3.2.js"></script>
<script src="js/fun.js"></script>
```

接着，创建 form 表单，添加表单元素，完成查询关键字的提交。关键代码如下：

```
<form name="form1" method="post" search_title="index.php">
<tr>
    <td height=35 align="center" bgcolor="#CCCCCC"><span class="STYLE_select5">请输入标题名称：
</span>
        <input name="search_title" type="text" id="search_title">
        <input type="submit" name="Submit" value="提交">
        <a href="Rss.xml" class="STYLE3">查看 XML 文件内容</a>
    </td>
```

```
</tr>
</form>
```

最后，实例化 Search_XML 类，执行 Select_XML()方法，将表单中提交的关键字作为参数传递到 Select_XML()方法中，完成对 XML 的查询。关键代码如下：

```
<?php
    if(isset($_POST['Submit']) && isset($_POST['search_title']) && $_POST['search_title']!=""){
                                                    //判断查询提交按钮是否为真
        $Search_XML = new Search_XML;               //实例化类
        $Search_XML->Select_XML($_POST['search_title']);
            //执行类中方法，并对提交的关键字进行编码转换
    }else{    //如果提交按钮和查询关键字为空，则输出如下内容
    ?>
    <div>
      <p> </p>
    </div>
    <?php
    }
?>
```

（3）在实例的根目录中创建一个 Rss.xml 文件，作为查询的内容。

照猫画虎：上述实例采用的是 utf-8 编码格式，将编码格式修改为 gb2312，重新完成本实例。提示：需要应用 iconv()对编码格式进行转换。（20 分）（实例位置：光盘\mr\15\zmhh\03_zmhh）

15.5.4　基本功训练 4——动态更新 XML 中的数据

视频讲解：光盘\mr\15\lx\动态更新 XML 中的数据.exe

实例位置：光盘\mr\15\zmhh\04

在 PHP 中修改 XML 文件，需要综合运用读取和查询 XML 文件的技术。先读取 XML 文件中的数据，然后根据超链接中传递的标题，通过查询定位到指定的 XML 文件，最终实现对 XML 文件的修改。运行结果如图 15.10 所示。

修改XML文件	
标题	如何通过PHP创建xml文件
链接地址	http://192.168.1.59/mr/15/zmhh/01_diy/rss.php?lmbs=6
内容摘要	详细内容请参考吉林省明日科技有限公司出版的典型模块一书。
修改数据	

图 15.10　PHP 修改 XML 文件

应用 DomDocument 类中的 createTextNode()和 replaceChild()方法完成对子节点值的修改。

（1）createTextNode()方法用来创建一个新的文本节点。语法如下：

```
class DomDocument {
    DOMText createTextNode ( string content )
}
```



参数 content 是文本的内容。成功则返回一个文本节点，失败则返回一个错误信息。

说明：TextNode()方法创建的文本节点不会出现在文档中，除非使用 DOMNode->appendChild()方法将其添加到文档中。

（2）replaceChild()方法用来修改一个子节点。语法如下：

```
class DOMNode {
    DOMNode replaceChild ( DOMNode newnode, DOMNode oldnode )
};
```

该方法实现将当前的 oldnode 节点替换为 newnode 节点。

在执行 XML 文件的修改操作之前，首先要读取 XML 文件中的数据，然后执行查询操作定位到指定节点，才可以执行修改的操作。

本实例中的操作是在 index.php 文件中完成的。具体步骤如下：

（1）完成类的创建和方法的定义。创建 Amend_XML 类，继承 DOM 类库。其中定义了 4 个方法，分别是构造函数，完成对指定 XML 文件的读取；show_amend()方法读取并输出 XML 文件中的数据；search_xml()方法执行查询操作，并将查询结果输出到 form 表单中，作为要修改的内容；amend_xml()方法完成对指定节点元素的修改操作。关键代码如下：

```
<?php
    class Amend_XML extends DomDocument{                 //定义类文件，继承 DOM 类对象
    public function __construct() {                      //构造函数
        parent:: __construct();
        $this->load("Rss.xml");                          //读取存储留言信息的 XML 文档 Rss.xml
    }
//省略了部分代码
public function amend_xml($title,$link,$description){    //定义保存数据的方法
    $xpath = new DOMXPath($this);                        //查询待修改的记录
    $Node_Record    = $xpath->query("//item[title=$title]");  //执行查询操作，同样以标题为依据
    $Replace[0]=$title;
    $Replace[1]=$link;
    $Replace[2]=$description;
    $d=0;
    foreach ($Node_Record->item(0)->childNodes as $articles) {
        $Node_newText = $this->createTextNode($Replace[$d]);          //对提交的修改数据进行转码
        $articles->replaceChild($Node_newText,$articles->lastChild);  //执行修改操作
        $d++;
    }
    echo "<script>alert('修改成功');location.href='".$_SERVER['PHP_SELF']."'</script>";
    $this->save("Rss.xml");
}
}
?>
```

（2）实例化 Amend_XML 类，应用 switch 语句，根据参数传递的不同值执行不同的方法，完成对 XML 文件的各种操作。关键代码如下：

```
<?php
    $amend_xml = new Amend_XML;                          //实例化类
    $Action ="";                                         //定义变量
    if(isset($_GET['Action']))                           //判断超链接传递的值
```

```
            $Action = $_GET['Action'];                           //获取超链接传递的值
            switch($Action){                                     //根据超链接传递的值进行判断
                case "show_amend":                               //浏览
                    $amend_xml->show_amend();
                break;
                case "search_xml":                               //查询
                    $amend_xml->search_xml($_GET['title']);
                break;
                case "amend_xml":                                //修改
                    $amend_xml->amend_xml($_GET['title'],$_POST['link'],$_POST['description']);
                break;
                default:
                    $amend_xml->show_amend();
            }
?>
```

照猫画虎：同样如果将上述实例修改为 gb2312 编码，必须应用 iconv()函数对字符串的编码格式进行转换，因为在页面中使用的是 gb2312 编码，而在 DOM 类库中方法采用的是 utf-8 编码格式。（20 分）（实例位置：光盘\mr\15\zmhh\04_zmhh）

15.5.5 基本功训练 5——动态删除 XML 中的数据

视频讲解：光盘\mr\15\lx\动态删除 XML 中的数据.exe

实例位置：光盘\mr\15\zmhh\05

在 PHP 中可以动态创建、查询、修改 XML 文件，同样可以动态删除 XML 文件中的数据。运行本实例，结果如图 15.11 所示。单击页面中的“删除”超链接，即可删除指定的 XML 数据。

删除XML文件		
标题：	如何使用Rss阅读器	删除
如何使用Rss阅读器如何使用Rss阅读器如何使用Rss阅读器		
URL:	http://192.168.1.59/mr/15/zmhh/01_diy/rss.php?lmbs=9	

图 15.11　PHP 删除 XML 文件

应用 DOM 类库中的 removeChild()方法删除 XML 文档中指定子节点。removeChild()方法的语法如下：

```
class DOMNode {
    DOMNode removeChild ( DOMNode oldnode )
}
```

参数 oldnode 为被删除的子节点。成功则返回被删除的节点。删除成功后，需要通过 save()方法重新保存文件。

在本实例中主要应用到 3 个自定义的方法：第一个是构造函数，完成读取指定 XML 文件的操作；第二个是 delete_xml()方法，执行 XML 的删除操作；第三个是 show_xml()方法，读取出 XML 文档中的数据。具体步骤如下：

首先，在 index.php 文件中创建 Delete_XML 类，继承 DOM 类库，定义构造方法完成对 XML 的读取；定义 delete_xml()方法执行删除 XML 节点的操作；定义 show_xml()方法实现输出 XML 文档中的数据。关键代码如下：



```
<?php
class Delete_XML extends DomDocument{                       //定义类文件，继承 DOM
      private $Root;                                        //定义成员变量
      public function __construct() {                       //定义构造方法
            parent:: __construct();
            $this->load("Rss.xml");                         //读取 XML 文档 Rss.xml
      }
      public function delete_xml($title){             //定义删除 XML 节点的方法，以超链接传递的值为参数
            $Root=$this->documentElement;                              //定义根目录
            $xpath = new DOMXPath($this);                              //应用 XPath 方法查询数据
            $Node_Record= $xpath->query("//item[title='$title']");     //以标题为条件进行查询
            $Root->removeChild($Node_Record->item(0));                 //根据查询结果，执行删除操作
            $this->save("Rss.xml");                                    //重新保存文件
            echo "<script>alert('删除成功');location.href='".$_SERVER['PHP_SELF']."'</script>";
      }
      public function show_xml(){                                      //定义输出 XML 文档的方法
            $Node_Record = $this->getElementsByTagName("item");        //获取节点
            $Node_Record_Length   =$Node_Record->length;               //获取节点的个数
            for($i=0;$i<$Node_Record->length;$i++)  {                  //执行 for 循环
                  $K=0;
                  foreach ($Node_Record->item($i)->childNodes as $articles) {   //应用 foreach 语句输出数据
                        $field[$K]=$articles->textContent;
                        $K++;
                  }
            echo  '<table  width="672"  border="1"  cellpadding="1"  cellspacing="1"  bordercolor="#FFFFFF" 
bgcolor="#999999"><tr>';
            echo '<td width="102" height="22" align="right" bgcolor="#FFFFFF">标题：</td>';
            echo '<td width="368" bgcolor="#FFFFFF"> '.$field[0].'</td>';
            echo '<td width="184" align="center" bgcolor="#FFFFFF"><a href="?Action=delete_xml&title='.$field
[0].'">删除</a></td>';
            echo '</tr><tr>';
            echo  '<td  colspan="3"  height="40"  style="padding-left:5px;  padding-right:5px;  padding-top:5px; 
padding-bottom:5px;" bgcolor="#FFFFFF">  '.$field[2].'</td>';
            echo '</tr><tr>';
           echo '<td height="25" align="right" bgcolor="#FFFFFF">URL:</td>';
            echo '<td colspan="2" bgcolor="#FFFFFF">  '.$field[1].'</td>';
            echo '</tr></table>';
           echo '</table>';
            }
      }
}
?>
```

然后，实例化 Delete_XML，应用 switch 语句，根据传递的不同参数，分别执行不同的方法，完成对 XML 的删除和输出操作。关键代码如下：

```
<?php
      $Action ="";                                        //定义变量
      $DeleteXML = new Delete_XML;                        //实例化删除类
      if(isset($_GET['Action']))                          //判断超链接传递的值
            $Action = $_GET['Action'];                    //获取超链接传递的值
            switch($Action){                              //根据超链接传递的值进行判断
```

```
                case "delete_xml"://删除
                    $DeleteXML->delete_xml($_GET['title']);        //执行删除操作
                break;
                default:
                $DeleteXML->show_xml();
        }
?>
```

最后在根目录下创建 Rss.xml 文件，定义被操作的内容。

照猫画虎：整合应用 PHP 动态操作 XML 技术，采用 gb2312 编码格式，开发一个 PHP 与 XML 的留言板。（20 分）（实例位置：光盘\mr\15\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

15.6　情景应用——拓展与实践

在了解了 RSS 2.0 的标准格式和创建了支持 RSS 阅读的站点之后，下面在情景应用中开发一个简单的 RSS 阅读器。

15.6.1　情景应用 1——创建 RSS 阅读器的框架

视频讲解：光盘\mr\15\lx\创建 RSS 阅读器的框架.exe

实例位置：光盘\mr\15\qjyy\01

RSS 阅读器的页面通过 frame 框架来完成，包括两部分：一部分是频道组和频道的树状导航菜单，另一部分是内容展示，运行结果如图 15.12 所示。

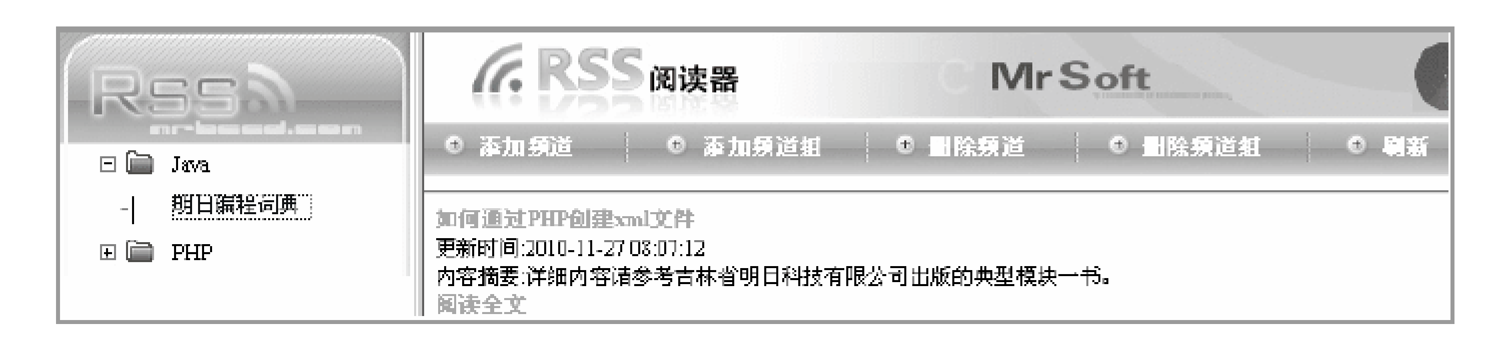


图 15.12　RSS 阅读器的框架

具体实现步骤如下：

（1）在本实例中应用到 Smarty 模板技术实现网页的动静分离，应用 PDO 抽象层完成对数据库的操作。其具体方法都封装于 system.class.inc.php、system.inc.php 和 system.smarty.inc.php 文件中，并存储于 system 文件夹下，有关这两项内容读者可以参考本书第 17 堂课和第 16 堂课，这里只是应用此技术，不对它们进行系统的讲解。

（2）创建首页的 index.php 文件和 index.tpl 模板文件，在 index.php 中包含类的实例化文件，并且通过 Smarty 中的 display()方法指定模板页。关键代码如下：



```
<?php
```



```
header('Content-Type: text/html; charset=utf-8' );                    //设置页面编码格式
require_once("system/system.inc.php");                                //包含类的实例化文件
$smarty->display("index.tpl");                                        //指定模板页
?>
```

在 index.tpl 文件中应用 frame 框架技术，设计阅读器的框架结构，分割成两部分内容：左侧为阅读器频道组和频道的树状导航菜单（channel_left.php），右侧为对应内容的展示区（channel_content.php）。代码如下：

```
<frameset cols="200,795">
<frame src="channel_left.php" name="leftwindow">
<frame src="channel_content.php" name="mainwindow">
</frameset><noframes></noframes>
```

（3）创建 channel_left.php 和 channel_left.tpl 文件，完成阅读器频道组和频道内容的树状导航设计，其关键是应用 JS 脚本控制树状导航菜单的展开和隐藏操作，通过 Smarty 完成页面的动静分离，通过 PDO 操作 MySQL 数据库。channel_left.php 的关键代码如下：

```
<?php
header('Content-Type: text/html; charset=utf-8' );                    //设置页面编码
require_once("system/system.inc.php");                                //包含类实例化文件
$big_sql="select * from tb_channel_group ";                           //定义 SQL 语句
$rst = $admindb->ExecSQL($big_sql,$conn);                             //执行语句
$smarty->assign("channel_group",$rst);                                //将返回结果赋给指定变量
//输出小类
$small_sql="select * from tb_channel ";
$small_rst = $admindb->ExecSQL($small_sql,$conn);
$smarty->assign("channel",$small_rst);
$smarty->display("channel_left.tpl");                                 //指定模板页
?>
```

在 channel_left.tpl 文件中，通过 Smarty 中的 section 语句循环输出频道组和频道的内容，通过 JS 脚本控制频道组的隐藏和展开。

（4）创建 channel_content.php 和 channel_content.tpl 文件，完成阅读器内容展示区的设计。在 channel_content.tpl 模板文件中，为频道添加、频道删除、频道组添加和频道组删除创建超链接，定义 div 标签，通过 JS 脚本文件完成内容的载入操作。关键代码如下：

```
<script src="js/channel_conment.js"></script>
<script type="text/javascript" src="js/channel_link.js"></script>
<table width="795" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td colspan="5"><img src="images/bg_10-(3)_1.jpg" width="795" height="52"></td>
  </tr>
  <tr>
    <td><a href="#" onClick="MM_openBrWindow('insert_channel.php','','width=515,height=237')"><img src=
"images/bg_10-(3)_2.jpg" width="114" height="33" border="0"></a></td>
    <td><a href="#" onClick="MM_openBrWindow('insert_channel_group.php','','width=515,height=237')"><img src
="images/bg_10-(3)_3.jpg" width="136" height="33" border="0"></a></td>
    <td><a href="delete_channel.php" target="_blank"><img src="images/bg_10-(3)_4.jpg" width="117" height
="33" border="0"></a></td>
    <td><a href="delete_channel_group.php" target="_blank"><img src="images/bg_10-(3)_5.jpg" width="132"
height="33" border="0"></a></td>
    <td width="296" background="images/bg_10-(3)_6.jpg"><a href="#" target="mainwindow" onClick="return
```

```
Refresh();" >
      <div  id="refresh1"  style="display:none;"><img  src="images/bg_10-(3)_6.jpg"  width="296"  height="33"
border="0"></div></a></td>
   </tr>
</table>
<p><div id="xmlpage" class="STYLE12"></div></p>
<div class="STYLE12" id="xml_url" style="display:none;"></div>
```

DIY：通过 frame 框架创建一个聊天室的框架结构。（20 分）（实例位置：光盘\mr\15\qjyy\01_diy）

15.6.2　情景应用 2——树状导航菜单输出频道组和频道

视频讲解：光盘\mr\15\lx\树状导航菜单输出频道组和频道.exe

实例位置：光盘\mr\15\qjyy\02

在 RSS 阅读器中，创建的频道组和频道是通过树状导航菜单来输出的，运行结果如图 15.13 所示。



图 15.13　树状导航菜单输出频道组和频道

树状导航菜单的创建是通过 JavaScript 脚本和 Smarty 的内建函数共同完成的。其关键是编写自定义脚本函数，控制频道组的展开和隐藏。自定义函数 open_close()的语法如下：

```
function open_close(x,y){
        if(document.getElementById(x).style.display==""){
            document.getElementById(x).style.display="none";
              document.getElementById(y).src="images/bg_10 (1).jpg";
        }else if( document.getElementById(x).style.display=="none"){
            document.getElementById(x).style.display="";
              document.getElementById(y).src="images/bg_10 (2).jpg";
        }
 }
```

通过自定义脚本函数，判断表格的样式是空还是 none。当样式 style.display 的值是空时，为 style.display 赋值为 none，并且输出图片 bg_10(1).jpg；当样式 style.display 的值是 none 时，为 style.display 赋值为空，并且输出图片 bg_10(2).jpg。

在 Smarty 模板页中通过树状导航菜单输出频道组和频道的内容。channel_left.tpl 的关键代码如下：

```
{php}$i=0;{/php}<!--定义一个变量，用于设置表格的 ID-->
<!--循环读取动态页中模板变量中的数据，输出频道组的数据-->
{section name=channel_group_id loop=$channel_group}
{php}$i++;{/php}        <!--增加 ID 值-->
<table width="85%" border="0" cellspacing="0" cellpadding="0">
       <tr>
              <!--根据单元格的 ID 值，输出不同的图片-->
              <td width="30%"><img src="images/bg_10 (1).jpg" id="images{php}echo $i;{/php}"></td>
              <!--应用 onClick 事件调用自定义函数，实现树状菜单的折叠和展开-->
              <td width="70%" align="left" height="24" onClick="javascript:open_close('id{php}echo $i;{/php}','images
```

```
{php}echo $i;{/php}');" >{$channel_group[channel_group_id].tb_channel_group_name}</td>
        </tr>
</table>
<!--定义表格的 ID 值，设置表格的样式-->
<table width="85%" border="0" cellspacing="0" cellpadding="0" id="id{php}echo $i;{/php}" style="display:none;">
<!--通过 section 函数，循环输出指定频道组中频道的数据-->
{section name=channel_id loop=$channel}
{if $channel_group[channel_group_id].tb_channel_group_id==$channel[channel_id].tb_channel_group}
      <tr>
            <td width="30%" align="center">-|</td>
            <!--设置超链接，通过 onClick 调用指定的函数，实现频道内容的输出-->
            <td height="24" width="70%" align="left">
                  <a href="channel_content.php" target="mainwindow" onClick="return ReqXml('{$channel[channel_
id].tb_channel_address}');">{$channel[channel_id].tb_channel_name}</a>
            </td>
      </tr>
{/if}
{/section}
</table>
{/section}
```

DIY：根据上述 JS 脚本，开发一个收缩式导航菜单。提示：利用显示隐藏表格来实现收缩式导航菜单的功能。单击“导航”超链接，显示当前菜单的内容，隐藏上一个显示的菜单。(20 分)(实例位置：光盘\mr\15\qjyy\02_diy)

15.6.3　情景应用 3——Ajax 无刷新读取订阅信息

视频讲解：光盘\mr\15\lx\Ajax 无刷新读取订阅信息.exe

实例位置：光盘\mr\15\qjyy\03

在 RSS 阅读器中，读取指定 RSS 文件中的订阅信息应用的是 Ajax 无刷新技术，最后通过 div 标签来完成订阅信息的输出。

1. 通过自定义函数 ReqXml()读取 RSS 文件

读取指定的 RSS 文件应用自定义函数 ReqXml()实现，并存储于 js\channel_comment.js 文件中。语法如下：

```
function ReqXml(url){
top.mainwindow.xml_url.innerHTML=url;
top.mainwindow.xmlpage.innerHTML="";
top.mainwindow.refresh1.style.display="";
      CreateXMLHttpRequest();
      xmlobj.onreadystatechange=StatHandler; //定义获取服务器端相应的方法
      xmlobj.open("get",url,true);              //执行该文件
      xmlobj.send(null);                        //向服务器发送请求
return false;
}
```

在 RSS 阅读器的 channel_left.tpl 文件中，通过 onClick 事件调用 ReqXml()函数，并且将数据库中存储的 RSS 文件的地址作为参数。channel_left.tpl 的关键代码如下：

```
<a href="channel_content.php" target="mainwindow"
      onClick="return ReqXml('{$channel[channel_id].tb_channel_address}');">
      {$channel[channel_id].tb_channel_name}
</a>
```

2．通过自定义函数 StatHandler()输出订阅内容

在收到服务器端的消息时，自动执行 StatHandler()函数，获取服务器端发送的 XML 消息。读取 XML 文档中 item 元素的数据，应用 for 循环输出 item 元素中的数据，并重新定义数据输出的格式，最后使用从服务器端发来的消息更新客户端的显示内容。StatHandler()函数存储于 js\channel_conment.js 文件中，代码如下：

```
function StatHandler(){
      if(xmlobj.readystate==4 && xmlobj.status==200){
            xml=xmlobj.responseXML;                              //获取服务器端发来的 xml 信息
            var allees=xml.getElementsByTagName("item");         //获取 xml 文档中的 item 元素
            row="";
            for(var i=0; i<allees.length;i++){                   //执行 for 循环
                  //定义变量
                  var tempobj,item_title,item_link,item_pubDate,item_description,item_links;
                  tempobj=allees[i].getElementsByTagName("link");     //获取 link 元素的值
                  item_link=tempobj[0].childNodes[0].nodeValue;       //将值赋给 item_link
                  tempobj=allees[i].getElementsByTagName("title");    //获取 title 的值
                  item_title="  <a                             href='"+item_link+"'"+"target='_blank'"+"
class='ss'>"+tempobj[0].childNodes[0].nodeValue+"</a>"+"<br>";      //设置标题的输出内容
                  item_link="  <a  href='"+item_link+"'"+"target='_blank'"+"  class='ss'>"+" 阅 读 全 文
"+"</a>"+"<br>";
                  tempobj=allees[i].getElementsByTagName("pubDate");    //获取创建时间的值
                  item_pubDate="  更新时间:"+tempobj[0].childNodes[0].nodeValue+"<br>";
                  tempobj=allees[i].getElementsByTagName("description");
                  counts=tempobj.length;
                  if(counts>0){                                       //判断当存在内容摘要时，输出下面的内容
                        item_description="  内容摘要:"+tempobj[0].childNodes[0].nodeValue+"<br>";
                  }else{                                              //如果没有内容摘要则输出下面的内容
                        item_description="  没有内容摘要！"+"<br>";
                  }
            row=item_title+item_pubDate+item_description+item_link+"<HR>";
                  top.mainwindow.xmlpage.innerHTML+=row;
            }
      }
}
```

3．刷新订阅信息

所谓刷新订阅信息，就是应用自定义函数 Refresh()，重新执行一次无刷新读取订阅信息的操作。其存储于 js\channel_conment.js 文件中，语法如下：

```
function Refresh(){
xml_ss=xml_url.innerText;
ReqXml(xml_ss);
return false;
}
```

在设置的刷新超链接中，应用 onClick 事件调用 Refresh()函数，重新执行一次 ReqXml()函数，实现重新读取和输出 RSS 文件中的内容。刷新超链接在 channel_comment.tpl 文件中定义，其代码如下：

```
<a href="#" target="mainwindow" onClick="return Refresh();" >
    <div id="refresh1" style="display:none;"><img src="images/bg_10-(3)_6.jpg" width="296" height="33" border="0"></div></a>
```

DIY：通过 JQuery 读取 XML 中的数据。（20 分）（实例位置：光盘\mr\15\qjyy\03_diy）

15.6.4　情景应用 4——添加频道组

视频讲解：光盘\mr\15\lx\添加频道组.exe

实例位置：光盘\mr\15\qjyy\04

要想通过 RSS 阅读器来订阅信息，保证订阅信息的条理清晰、查找方便，最好的方法就是将其进行分类处理。创建不同类别的频道组，将订阅信息按照类别放在不同的频道组中，并且在同一频道组中创建不同的频道，以区分不同订阅信息的内容。下面将介绍 RSS 阅读器中频道组的创建方法，运行结果如图 15.14 所示。



图 15.14　频道组的创建

在 RSS 阅读器模块中应用的是 Smarty 模板技术，所以频道组的添加主要由 3 个文件组成：insert_channel_group.php 用于指定 Smarty 的模板页；insert_channel_group.tpl 是模板页，用于创建添加频道组的 form 表单和页面，并且通过 Ajax 实现频道组的无刷新添加；insert_channel_group_ok.php 用于将表单中提交的数据存储到数据库中，完成频道组的添加。操作步骤如下：

（1）创建 insert_channel_group.php 文件，用于指定 Smarty 的模板页。代码如下：

```
<?php
header('Content-Type: text/html; charset=utf-8' );     //设置页面编码格式
require_once("system/system.inc.php");                 //包含类的实例化文件
$smarty->display("insert_channel_group.tpl");          //指定模板页
?>
```

（2）创建 insert_channel_group.tpl 模板页，添加 form 表单，调用 channel_group()函数，在服务器端执行 insert_channel_group_ok.php 文件，实现频道组的无刷新添加，通过 handleServerResponse()函数返回从服务器端获取的信息。channel_group()函数存储于 js\channel_group.js 文件中，其语法如下：

```
function channel_group(){
    if(form1.channel_group_name.value==""){
        alert("请输入频道组名称！");
            form1.channel_group_name.select();
             return(false);
        }
    if(xmlHttp.readyState==4 || xmlHttp.readyState==0){ //在 xmlHttp 对象不忙时进行处理
```

```
        names = document.getElementById("channel_group_name").value;  //获取频道组名称
    //在服务器端执行 insert_channel_group_ok.php
    xmlHttp.open("GET","insert_channel_group_ok.php?channel_group_names="+names,true);
    xmlHttp.onreadystatechange=handleServerResponse;    //定义获取服务器端响应的方法
    xmlHttp.send(null);   //向服务器发送请求
    return false;
    }else
        setTimeout('channel_group()',1000);                //如果服务器忙，1 秒后重试
}
```

（3）创建 insert_channel_group_ok.php 文件，将创建的频道组名称添加到数据库中。代码如下：

```
<?php
header('Content-Type: text/html; charset=utf-8' );
require_once("system/system.inc.php");
$sql="insert into tb_channel_group(tb_channel_group_name)values('".iconv("gbk","utf-8",$_GET['channel_ group_names']).")";
$rst = $admindb->ExecSQL($sql,$conn);                         //执行语句
if($rst){
    echo "频道组添加成功!";
}else{
    echo "频道组添加失败!";
}
?>
```

DIY：上述内容实现频道组的添加操作，在此基础上完成频道组的删除操作。（20 分）（实例位置：光盘\mr\15\qjyy\04_diy）

15.6.5　情景应用 5——添加频道

视频讲解：光盘\mr\15\lx\添加频道.exe

实例位置：光盘\mr\15\qjyy\05

在 RSS 阅读器模块中，频道中的内容是存储在对应的频道组下的，这里将讲解如何实现将频道内容添加到对应的频道组下，运行结果如图 15.15 所示。

频道的添加实质是将指定的 RSS 文件的地址存储到数据库中，并且对 RSS 文件的有效性进行判断。操作步骤如下：

（1）创建 insert_channel.php 文件，指定模板页 insert_channel.tpl。在 insert_channel.tpl 模板页中，应用 clipboardData 对象中的 getData 方法获取剪贴板中的 RSS 文件的地址，并且将订阅地址通过 form 表单提交到 insert_channel_ok.php 页中，继续执行频道添加的操作，运行结果如图 15.16 所示。

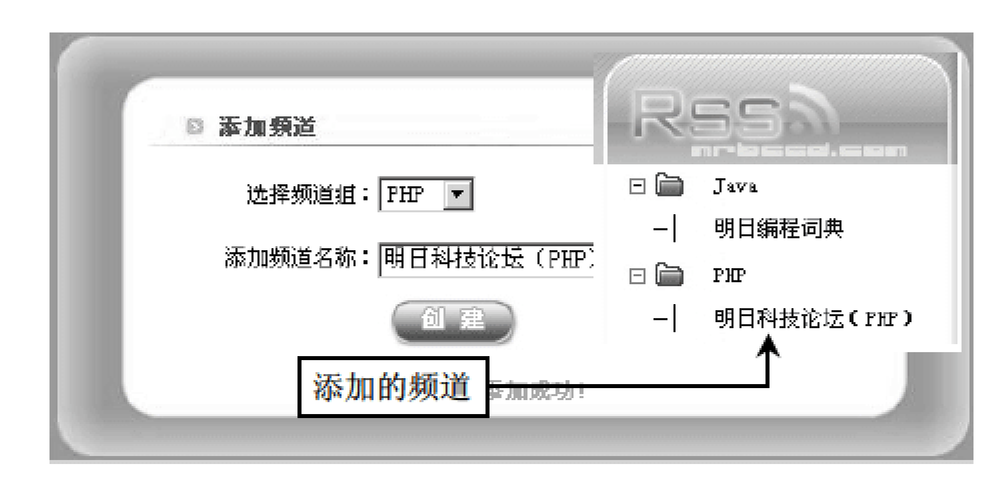


图 15.15　添加频道

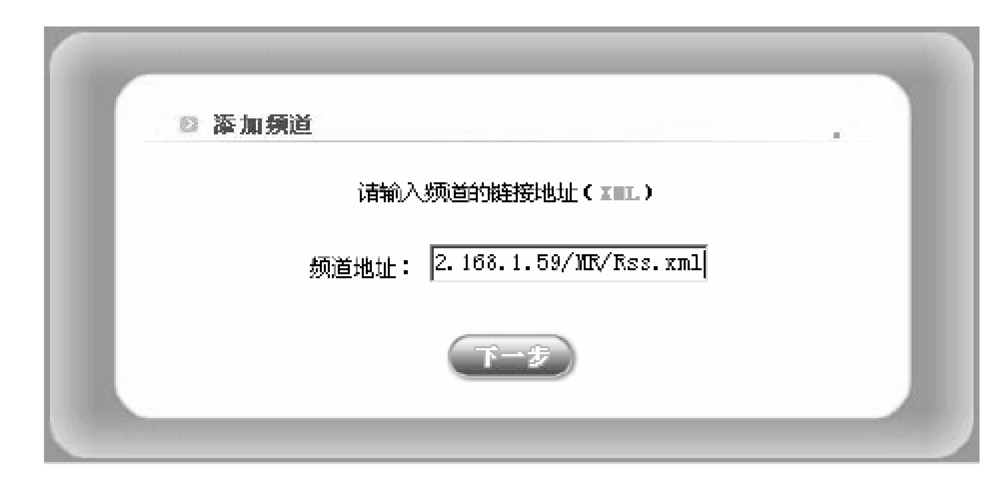


图 15.16　添加频道第一步，获取订阅地址



insert_channel.tpl 的关键代码如下：

```
<script type="text/javascript" src="js/channel.js"></script><!--通过脚本判断 RSS 文件的地址是否合理-->
<!--通过 clipboardData 对象的 getData 方法获取剪贴板中的地址，并添加到文本框中 -->
<body onLoad="form1.tb_channel_address.value=window.clipboardData.getData('Text')">
<form name="form1" method="post" action="insert_channel_ok.php">
<table width="515" height="237" border="0" cellpadding="0" cellspacing="0" background="images/添加频道_3.jpg">
  <tr>
    <td align="center" class="STYLE12">
      <p>请输入频道的链接地址（<span class="STYLE1">XML</span>）</p>
      <p>频道地址：<input type="text" name="tb_channel_address"></p></td>
  </tr>
  <tr>
    <td align="center" valign="top"><input type="image" name="imageField" src="images/添加频道_4.jpg" onClick="return channel_address()"></td>
  </tr>
  </table>
</form>
```

（2）创建 insert_channel_ok.php 文件，判断添加的订阅地址是否存在，判断该地址是否已经被订阅。从数据库中读取频道组的数据，并将频道组的数据写入到一个模板变量中，通过模板变量将数据传递到模板页中；最后指定 insert_channel_ok.php 文件的模板页为 insert_channel_ok.tpl。关键代码如下：

```
<?php
header('Content-Type: text/html; charset=utf-8' );                    //设置编码格式
require_once("system/system.inc.php");                                //包含类实例化文件
$fp=@fopen($_POST['tb_channel_address'],"r");                         //判断指定的文件是否存在
fclose($fp);                                                          //关闭链接
if($fp==false) {
	echo "<script>alert('该文件不存在！'); history.back();</script>";
}else{
	$sqls="select * from tb_channel where  tb_channel_address='".$_POST['tb_channel_address']."'";
	$rst = $admindb->ExecSQL($sqls,$conn);                            //执行语句
	if($rst==false){                                                  //判断该频道是否已经被订阅
		$sql="select * from tb_channel_group ";                       //定义 SQl 语句
		$arrays = $admindb->ExecSQL($sql,$conn);                      //执行语句
		$smarty->assign("channel_groups",$arrays);                    //定义模板变量
		$smarty->assign("channel_group",1);
		$smarty->assign("tb_channel_address",$_POST['tb_channel_address']);
		$smarty->display("insert_channel_ok.tpl");                    //指定模板页
	}else{
		echo "<script>alert('该频道已经被订阅！'); history.back();</script>";
	}
}
?>
```

（3）在 insert_channel_ok.tpl 模板页中设置下拉列表框，将模板变量中传递的频道组的数据作为下拉列表的值，并设置文本框和隐藏域提交频道的名称和订阅地址，最后通过 onClick 事件调用 channel_check()函数实现频道信息的无刷新提交。关键代码如下：

```
<form name="form1" method="post" action="">
<table width="515" height="237" border="0" cellpadding="0" cellspacing="0">
	<tr>
```

```
<td width="184" align="right" class="STYLE12">选择频道组：</td>
            <td width="331">
<select name="channel_group">
       {section name=cha_id loop=$channel_group}
   <option
value="{$channel_group[cha_id].tb_channel_group_id}">{$channel_group[cha_id].tb_channel_group_name}</
option>
   {/section}
</select>
</td>
       </tr>
       <tr>
            <td align="right" class="STYLE12">添加频道名称：</td>
            <td><input type="text" name="tb_channel_name">
                 <input type="hidden" name="tb_channel_address" value="{$tb_channel_address}">
            </td>
       </tr>
       <tr>
                 <td  valign="top"><input  type="image"  name="imageField"  src="images/添加频道组_5.jpg"
onClick="return channel_check()"></td>
       </tr>
       <tr>
            <td height="60" colspan="2" align="center" valign="middle" class="STYLE1">
                 <div id="channel_check"></div>  </td>
       </tr>
</table>
</form>
```

（4）创建 insert_channels_ok.php 文件，将表单中提交的数据存储到数据库中。程序代码如下：

```
<?php
header('Content-Type: text/html; charset=utf-8' );
require_once("system/system.inc.php");
$channel_name=iconv("gbk","utf-8",$_GET['channel_name']);
$sqls="insert into tb_channel(tb_channel_name,tb_channel_address,tb_channel_group)values('".$channel_name."',
'".iconv("gbk","utf-8",$_GET['channel_address'])."','".iconv("gbk","utf-8",$_GET['channel_groups'])."')";
$rst = $admindb->ExecSQL($sqls,$conn);                         //执行语句
if($rst){
       echo "频道添加成功!";
}else{
       echo "频道添加失败!";
}
?>
```

DIY：上述内容实现频道的添加操作，在此基础上完成频道的删除操作。(20 分)(实例位置：光盘\mr\15\qjyy\05_diy)

情景应用 DIY 栏目分数统计：

<table>
<tr><td>DIY 题目</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td rowspan="2">总分数</td><td rowspan="2"></td></tr>
<tr><td>分数</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>



15.7　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．下面不属于 XML 特点的是（　　）。

A．易用性　　　B．结构性　　　C．紧凑性

2．XML 声明在文档中只能出现（　　）。

A．一次　　　B．两次　　　C．多次

3．XML 文档中使用的注释格式为（　　）。

A．//　　　B．<!---->　　　C．/**/

4．<表示的内容为（　　）。

A．<　　　B．>　　　C．=

5．&表示的内容为（　　）。

A．<　　　B．>　　　C．&

二、填空题（每题 10 分，5 道题）

1．创建一个对象句柄为$dom 的 DOM 对象代码为（　　）。

2．XML 将指定的文件解析到内存中需要使用函数（　　）。

3．XML 将创建的字符串解析到内存当中需要使用函数（　　）。

4．XML 将一个使用 dom 函数创建的 DomDocument 对象导入到内存当中需要使用函数（　　）。

5．XML 中创建一个新的文本节点需要使用方法（　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

15.8　行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

<table>
<tr><th>序号</th><th>内　容</th><th colspan="2">行 动 指 南</th></tr>
<tr><td rowspan="6">1</td><td rowspan="3">照猫画虎栏目
分数（　　）</td><td>分数>75 分</td><td>优秀，基本功掌握得不错，加油！</td></tr>
<tr><td>75 分>分数>50 分</td><td>及格，知识掌握得不牢，重新做一遍照猫画虎。</td></tr>
<tr><td>分数<50 分</td><td>请使用光盘中提供的“实战能力测试系统”进行提高训练。</td></tr>
<tr><td rowspan="3">情景应用栏目
分数（　　）</td><td>分数>75 分</td><td>优秀，综合应用能力很强。</td></tr>
<tr><td>75 分>分数>50 分</td><td>及格，综合应用能力需提高，再练一遍情景应用。</td></tr>
<tr><td>分数<50 分</td><td>请使用光盘中提供的“实战能力测试系统”进行提高训练。</td></tr>
</table>

续表

序号	内　容	行 动 指 南	
1	自我测试栏目 分数（　　）	分数>75 分	优秀，有成为编程高手的潜质。
		分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）创建 XML 留言板。	
		（2）开发一个完整的支持 RSS 阅读器的论坛。	
		（3）编辑 RSS 阅读器，读取论坛中的数据。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

15.9　成功可以复制——中国网络游戏产业的领军人陈天桥

成就：创建盛大网络有限责任公司，并于 2004 年在美国纳斯达克成功上市。

创始公司：盛大网络有限责任公司

财富巅峰：2004 年陈天桥所持有的盛大网络股票市值就高达 50 亿元。

最高学历：本科

出生日期：1973 年 5 月

超级语录：“当你认准一个方向的时候全力以赴，只有专注的企业才能成功，多元化的企业可以存活但是很难成功。”

成功历程

1994 年陈天桥毕业于复旦大学经济学专业，毕业后的最初几年，先后在上海陆家嘴集团和金信证券公司工作。

1999 年，陈天桥以 50 万元启动资金和 20 名员工为基础，在上海市的一所民宅里创立了盛大网络有限责任公司，并推出网络虚拟社区“天堂归谷”。

2000 年，盛大网络获得了中华网 300 万美元的注资。于是很快，初出茅庐的盛大广泛涉足了网上互动



娱乐社区的开发经营、即时通信软件的开发和服务以及网上动画、漫画的开发经营，而这种多元化的经营使盛大网络进入了迷茫而无序的发展状态。在功成名就之后，陈天桥告诉前来取经的创业者：“当你认准一个方向的时候全力以赴，只有专注的企业才能成功，多元化的企业可以存活但是很难成功。”而这正是他的切肤之痛，盲目发展很快结出了恶果，盛大网络陷入困顿中。

在总结了自己的失败原因之后，陈天桥开始寻求改变。这时《传奇》进入了他的视野并把他深深吸引了，他立刻给中华网写了厚厚一叠项目建议书，并告知投资方这是一个非常好的商业领域，认为到年底不但能赚钱而且能赚大钱。但是投资方觉得陈天桥在讲一个神话，于是对他说“你可以一个人走，但是我们不陪你走。”最终，陈天桥与中华网分手，中华网按股份留给陈天桥 30 万美元。



2001 年 7 月 14 日，盛大和《传奇》海外版权持有商 Actoz 以每年 30 万美元的价格签约。陈天桥把剩余的 30 万美元全部交给了 Actoz。合同签完后，盛大资金就不足了，但游戏运营才刚开始，光服务器和网络带宽就需要一大笔钱，形势十分危险。为了解决硬件设备问题，他就拿着与韩国方面签订的合约，找到商家，告诉他们盛大要运作韩国人的游戏，申请试用机器两个月。对方一看是国际正规合同，于是就同意了。然后，拿着服务器的合约，以同样的方式找通信运营商谈，通信运营商最终给了盛大两个月测试期免费的带宽试用。有了韩方的合同，再加上服务器厂家和中国电信的支持，陈天桥又取得了游戏分销商的信任，代销盛大游戏点卡，分成 33%。

2001 年 9 月 28 日，《传奇》开始公测，两个月后正式收费，同时在线人数迅速突破 40 万大关，全国点卡集体告罄，资金迅速回笼，盛大安然渡过了这场生死玄关。这时候离盛大和 Actoz 签约仅仅 4 个月。

2004 年，专注于游戏产业而获得巨额财富的盛大在纳斯达克上市，陈天桥所持有的盛大网络股票市值就高达 50 亿元。时至今日，陈天桥一直把游戏产业作为盛大公司的主营业务，一举成为中国网络游戏业的领头羊。

深度评价

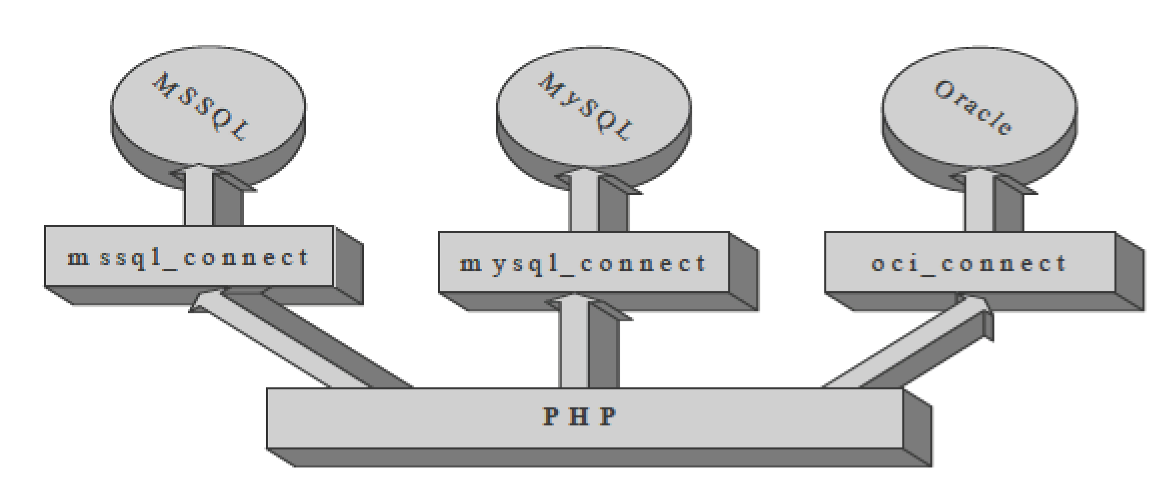
从陈天桥的成功历程，我们可以看出专注的重要性，由于他一时的疏忽，在企业实力还不是很强大的情况下涉足过多的产业，差一点断送了盛大的“前途”。后来盛大专注于网络游戏产业的经营，终于走出困境而获得巨大成功，所以我们学习编程也一样，不要三心二意，今天想把 C#学好，过几天又想把 VB 学好，这样只能是一事无成，最终哪个也没有学精。

第16堂课

PDO 数据库抽象层

（视频讲解：82 分钟）

在 PHP 的早期版本中，各种不同的数据库扩展（MySQL、MS SQL、Oracle）根本没有真正的一致性，虽然都可以实现相同的功能，但是这些扩展却互不兼容，都有各自的操作函数，各自为政，如下图所示。结果导致 PHP 的维护非常困难，可移植性也非常差，为了解决这些问题，PHP 的开发人员编写了一种轻型、便利的 API 来统一各种数据库的共性，从而达到 PHP 脚本最大程度的抽象性和兼容性，这就是数据库抽象层。而在本堂课中将要介绍的是目前 PHP 抽象层中最为流行的一种——PDO 抽象层。



学习摘要：

- 了解 PDO
- PDO 连接数据库
- PDO 中执行 SQL 语句
- PDO 中获取结果集
- PDO 中捕获 SQL 语句中的错误
- PDO 中错误处理
- PDO 中事务处理
- PDO 中存储过程

16.1　什么是 PDO

16.1.1　PDO 概述

PDO 是 PHP Date Object（PHP 数据对象）的简称，它是与 PHP 5.1 版本一起发行的，目前支持的数据库包括 Firebird、FreeTDS、Interbase、MySQL、MS SQL Server、ODBC、Oracle、Postgre SQL、SQLite 和 Sybase。有了 PDO，不必再使用 mysql_*函数、oci_*函数或者 mssql_*函数，也不必再为它们封装数据库操作类，只需要使用 PDO 接口中的方法就可以对数据库进行操作。在选择不同的数据库时，只需修改 PDO 的 DSN（数据源名称）。

在 PHP 6 中将默认使用 PDO 连接数据库，所有非 PDO 扩展将会在 PHP 6 中被移除。该扩展提供 PHP 内置类 PDO 来对数据库进行访问，不同数据库使用相同的方法名解决数据库连接不统一的问题。

16.1.2　PDO 特点

PDO 是一个"数据库访问抽象层"，作用是统一各种数据库的访问接口，与 MySQL 和 MSSQL 函数库相比，PDO 让跨数据库的使用更具有亲和力；与 ADODB 和 MDB2 相比，PDO 更高效。

PDO 将通过一种轻型、清晰、方便的函数，统一各种不同 RDBMS 库的共有特性，实现 PHP 脚本最大程度的抽象性和兼容性。

PDO 吸取现有数据库扩展成功和失败的经验教训，利用 PHP 5 的最新特性，可以轻松地与各种数据库进行交互。

PDO 扩展是模块化的，使用户能够在运行时为数据库后端加载驱动程序，而不必重新编译或重新安装整个 PHP 程序。例如，PDO_MySQL 扩展会替代 PDO 扩展实现 MySQL 数据库 API。还有一些用于 Oracle、PostgreSQL、ODBC 和 Firebird 的驱动程序，更多的驱动程序尚在开发。

16.1.3　安装 PDO

PDO 是与 PHP 5.1 一起发行的，默认包含在 PHP 5.1 中。由于 PDO 需要 PHP 5 核心面向对象特性的支持，因此其无法在 PHP 5.0 之前的版本中使用。

默认情况下，PDO 在 PHP 5.2 中为开启状态，但是要启用对某个数据库驱动程序的支持，仍需要进行相应的配置操作。

在 Linux 环境下使用 MySQL 数据库，可以在 configure 命令中添加如下选项：

```
--with-pdo-mysql=/path/to/mysql/installation
```

在 Windows 环境下，PDO 在 php.ini 文件中进行配置，如图 16.1 所示。



图 16.1　Windows 环境下配置 PDO

要启用 PDO，首先必须加载“extension=php_pdo.dll”，如果要想其支持某个具体的数据库，那么还要加载对应的数据库选项。例如，要支持 MySQL 数据库，则需要加载“extension=php_pdo_mysql.dll”选项。

注意：在完成数据库的加载后，要保存 php.ini 文件，并且重新启动 Apache 服务器，修改才能够生效。

16.2　PDO 连接数据库

16.2.1　PDO 构造函数

在 PDO 中，要建立与数据库的连接需要实例化 PDO 的构造函数。PDO 构造函数的语法如下：

```
__construct(string $dsn[,string $username[,string $password[,array $driver_options]]])
```

参数说明：

☑ dsn：数据源名，包括主机名端口号和数据库名称。
☑ username：连接数据库的用户名。
☑ password：连接数据库的密码。
☑ driver_options：连接数据库的其他选项。

通过 PDO 连接 MySQL 数据库的代码如下：

```php
<?php
	header("Content-Type:text/html;charset=utf-8");	//设置页面的编码格式
	$dbms='mysql';								//数据库类型
	$dbName='db_database16';					//使用的数据库名称
	$user='root';								//使用的数据库用户名
	$pwd='111';									//使用的数据库密码
	$host='localhost';							//使用的主机名称
	$dsn="$dbms:host=$host;dbname=$dbName";
	try {										//捕获异常
		$pdo=new PDO($dsn,$user,$pwd);			//实例化对象
		echo "PDO 连接 MySQL 成功";
	} catch (Exception $e) {
		echo $e->getMessage()."<br>";
	}
?>
```

16.2.2　DSN 详解

DSN 是 Data Source Name（数据源名称）的首字母缩写。DSN 提供连接数据库需要的信息。PDO 的 DSN 包括 3 部分，即 PDO 驱动名称（如 mysql、sqlite 或者 pgsql）、冒号和驱动特定的语法。每种数据库都有其特定的驱动语法。

在使用不同的数据库时，必须明确数据库服务器是完全独立于 PHP 的实体。虽然在讲解本书的内容时，数据库服务器和 Web 服务器是在同一台计算机上，但实际的情况可能不是如此。数据库服务器可能与 Web 服务器不是在同一台计算机上，此时要通过 PDO 连接数据库时，就需要修改 DSN 中的主机名称。

由于数据库服务器只在特定的端口上监听连接请求。每种数据库服务器具有一个默认的端口号（MySQL 是 3306），但是数据库管理员可以对端口号进行修改，因此有可能 PHP 找不到数据库的端口，此时就可以

在 DSN 中包含端口号。

另外由于一个数据库服务器中可能拥有多个数据库，所以在通过 DSN 连接数据库时，通常都包括数据库名称，这样可以确保连接的是想要的数据库，而不是其他人的数据库。

16.3 PDO 中执行 SQL 语句

在 PDO 中可以使用下面的 3 种方法来执行 SQL 语句。

16.3.1 exec()方法

exec()方法返回执行后受影响的行数。其语法如下：

```
int PDO::exec ( string statement )
```

参数 statement 是要执行的 SQL 语句。该方法返回执行查询时受影响的行数，通常用于 INSERT、DELETE 和 UPDATE 语句中。

16.3.2 query()方法

query()方法通过用于返回执行查询后的结果集。其语法如下：

```
PDOStatement PDO::query ( string statement )
```

参数 statement 是要执行的 SQL 语句。它返回的是一个 PDOStatement 对象。

16.3.3 预处理语句——prepare()和 execute()

预处理语句包括 prepare()和 execute()两个方法。首先，通过 prepare()方法做查询的准备工作，然后，通过 execute()方法执行查询。并且还可以通过 bindParam()方法来绑定参数提供给 execute()方法。其语法如下：

```
PDOStatement PDO::prepare ( string statement [, array driver_options] )
bool PDOStatement::execute ( [array input_parameters] )
```

你问我答：预处理语句。

它是要运行的 SQL 的一种编译过的模板，它可以使用变量参数进行定制。预处理语句可以带来两大好处：

（1）查询只需解析（或准备）一次，但是可以用相同或不同的参数执行多次。当查询准备好后，数据库将分析、编译和优化执行该查询的计划。对于复杂的查询，这个过程要花比较长的时间，如果需要以不同参数多次重复相同的查询，那么该过程将大大降低应用程序的速度。通过使用预处理语句，可以避免重复分析/编译/优化周期。简言之，预处理语句使用更少的资源，因而运行得更快。

（2）提供给预处理语句的参数不需要用引号括起来，驱动程序会处理这些。如果应用程序独占地使用预处理语句，那么可以确保没有 SQL 入侵发生。但是，如果仍然将查询的其他部分建立在不受信任的输入之上，那么就仍然存在风险。

你问我答：PDO 中执行 SQL 语句方法的选择。

（1）如果只是执行一次查询，那么 PDO->query 是较好的选择。虽然它无法自动转义发送给它的任何



数据，但是它在遍历 SELECT 语句的结果集方面是非常方便的。然而在使用这个方法时也要相当小心，因为如果没有在结果集中获取到所有数据，那么下次调用 pdo->query 时可能会失败。
（2）如果多次执行 SQL 语句，那么最理想的方法是 prepare 和 execute。这两个方法可以对提供给它们的参数进行自动转义，进而防止 SQL 注入攻击；同时由于在多次执行 SQL 语句时，应用的是预编译语句，还可以减少资源的占用，从而提高运行速度。

16.4　PDO 中获取结果集

在 PDO 中获取结果集有 3 种方法，即 fetch()、fetchAll()和 fetchColumn()。

16.4.1　fetch()方法

fetch()方法获取结果集中的下一行。其语法如下：

```
mixed PDOStatement::fetch ( [int fetch_style [, int cursor_orientation [, int cursor_offset]]] )
```

参数说明：

☑　fetch_style：控制结果集的返回方式，其可选值如表 16.1 所示。

表 16.1　fetch_style 控制结果集的可选值

值	说　明
PDO::FETCH_ASSOC	关联数组形式
PDO::FETCH_NUM	数字索引数组形式
PDO::FETCH_BOTH	两者数组形式都有，这是默认的
PDO::FETCH_OBJ	按照对象的形式，类似于以前的 mysql_fetch_object()
PDO::FETCH_BOUND	以布尔值的形式返回结果，同时将获取的列值赋给 bindParam()方法中指定的变量
PDO::FETCH_LAZY	以关联数组、数字索引数组和对象 3 种形式返回结果

☑　cursor_orientation：PDOStatement 对象的一个滚动游标，可用于获取指定的一行。

☑　cursor_offset：游标的偏移量。

例 16.01　通过 fecth()方法获取结果集中下一行的数据，进而应用 while 语句完成数据库中数据的循环输出，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.01）**

创建 index.php 文件，设计网页页面。首先，通过 PDO 连接 MySQL 数据库。然后，定义 SELECT 查询语句，应用 prepare()和 execute()方法执行查询操作。接着，通过 fetch()方法返回结果集中下一行数据，同时设置结果集以关联数组形式返回。最后，通过 while 语句完成数据的循环输出。关键代码如下：

```
<?php
$dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
$host='localhost';                          //数据库主机名
$dbName='db_database16';                    //使用的数据库
$user='root';                               //数据库连接用户名
$pass='111';                                //对应的密码
$dsn="$dbms:host=$host;dbname=$dbName";
try {
    $pdo = new PDO($dsn, $user, $pass);     //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
     $query="select * from tb_pdo_mysql";   //定义 SQL 语句
```

```
        $result=$pdo->prepare($query);              //准备查询语句
        $result->execute();                         //执行查询语句，并返回结果集
        while($res=$result->fetch(PDO::FETCH_ASSOC)){  //循环输出查询结果集，并且设置结果集为关联索引
        ?>
              <tr>
                <td height="22" align="center" valign="middle"><?php echo $res['id'];?></td>
                <td align="center" valign="middle"><?php echo $res['pdo_type'];?></td>
                <td align="center" valign="middle"><?php echo $res['database_name'];?></td>
                <td align="center" valign="middle"><?php echo $res['dates'];?></td>
                <td align="center" valign="middle"><a href="#">删除</a></td>
              </tr>
<?php
        }
                } catch (PDOException $e) {
       die ("Error!: " . $e->getMessage() . "<br/>");
}
                ?>
```

运行结果如图 16.2 所示。



图 16.2　fetch()方法获取查询结果集

16.4.2　fetchAll()方法

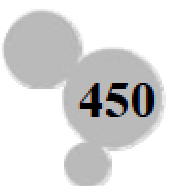
fetchAll()方法获取结果集中的所有行。其语法如下：

```
array PDOStatement::fetchAll ( [int fetch_style [, int column_index]] )
```

参数 fetch_style 控制结果集中数据的显示方式；参数 column_index 是字段的索引。其返回值是一个包含结果集中所有数据的二维数组。

例 16.02　通过 fecthAll()方法获取结果集中所有行，并且通过 for 语句读取二维数组中的数据，完成数据库中数据的循环输出，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.02）**

创建 index.php 文件，设计网页页面。首先，通过 PDO 连接 MySQL 数据库。然后，定义 SELECT 查询



语句，应用 prepare()和 execute()方法执行查询操作。接着，通过 fetchAll()方法返回结果集中所有行。最后，通过 for 语句完成结果集中所有数据的循环输出。关键代码如下：

```
<?php
$dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
$host='localhost';                                  //数据库主机名
$dbName='db_database16';                            //使用的数据库
$user='root';                                       //数据库连接用户名
$pass='111';                                        //对应的密码
$dsn="$dbms:host=$host;dbname=$dbName";
try {
    $pdo = new PDO($dsn, $user, $pass);             //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
     $query="select * from tb_pdo_mysql";           //定义 SQL 语句
     $result=$pdo->prepare($query);                 //准备查询语句
     $result->execute();                            //执行查询语句，并返回结果集
     $res=$result->fetchAll(PDO::FETCH_ASSOC); //获取结果集中的所有数据
     for($i=0;$i<count($res);$i++){                 //循环读取二维数组中的数据
     ?>
         <tr>
            <td height="22" align="center" valign="middle"><?php echo $res[$i]['id'];?></td>
            <td align="center" valign="middle"><?php echo $res[$i]['pdo_type'];?></td>
            <td align="center" valign="middle"><?php echo $res[$i]['database_name'];?></td>
            <td align="center" valign="middle"><?php echo $res[$i]['dates'];?></td>
            <td align="center" valign="middle"><a href="#">删除</a></td>
        </tr>
<?php
     }
} catch (PDOException $e) {
    die ("Error!: " . $e->getMessage() . "<br/>");
} ?>
```

运行结果如图 16.3 所示。



ID	PDO	数据库	时间	操作
1	pdo	mysql	2010-09-18 10:19:17	删除
2	pdo	Oracle	2010-09-18 10:19:17	删除
4	pdo	Access	2010-10-10 00:00:00	删除
22	pdo	SQL	2010-10-09 10:59:30	删除
23	pdo	db2	2010-10-16 10:59:41	删除

图 16.3　fetchAll()方法返回结果集中所有数据

16.4.3　fetchColumn()方法

fetchColumn()方法获取结果集中下一行指定列的值。其语法如下：

```
string PDOStatement::fetchColumn ( [int column_number] )
```

可选参数 column_number 设置行中列的索引值，该值从 0 开始。如果省略该参数则将从第一列开始取值。

例 16.03 通过 fecthColumn()方法获取结果集中下一行中指定列的值，注意这里是“结果集中下一行中指定列的值”。本实例输出数据表中第一列的值，即输出数据的 ID，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.03）**

创建 index.php 文件，设计网页页面。首先，通过 PDO 连接 MySQL 数据库。然后，定义 SELECT 查询语句，应用 prepare()和 execute()方法执行查询操作。接着，通过 fetchColumn()方法输出结果集中下一行第一列的值。关键代码如下：

```
<?php
$dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
$host='localhost';                              //数据库主机名
$dbName='db_database16';                        //使用的数据库
$user='root';                                   //数据库连接用户名
$pass='111';                                    //对应的密码
$dsn="$dbms:host=$host;dbname=$dbName";
try {
    $pdo = new PDO($dsn, $user, $pass);         //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
        $query="select * from tb_pdo_mysql";    //定义 SQL 语句
        $result=$pdo->prepare($query);          //准备查询语句
        $result->execute();                     //执行查询语句，并返回结果集
    ?>
        <tr>
          <td height="22" align="center" valign="middle"><?php echo $result->fetchColumn(0);?></td>
        </tr>
        <tr>
          <td height="22" align="center" valign="middle"><?php echo $result->fetchColumn(0);?></td>
        </tr>
        <tr>
          <td height="22" align="center" valign="middle"><?php echo $result->fetchColumn(0);?></td>
        </tr>
        <tr>
          <td height="22" align="center" valign="middle"><?php echo $result->fetchColumn(0);?></td>
        </tr>
<?php
          } catch (PDOException $e) {
    die ("Error!: " . $e->getMessage() . "<br/>");
}
          ?>
```

运行结果如图 16.4 所示。



图 16.4　fetchColumn()方法获取结果集中第一列的值



16.5　PDO 中捕获 SQL 语句中的错误

在 PDO 中捕获 SQL 语句中的错误有 3 种方案可以选择。

16.5.1　使用默认模式——PDO::ERRMODE_SILENT

在默认模式中设置 PDOStatement 对象的 errorCode 属性，但不进行其他任何操作。

例 16.04　通过 prepare 和 execute 方法向数据库中添加数据，设置 PDOStatement 对象的 errorCode 属性，手动检测代码中的错误，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.04）**

创建 index.php 文件，添加 form 表单，将表单元素提交到本页。通过 PDO 连接 MySQL 数据库，通过预处理语句 prepare()和 execute()执行 INSERT 添加语句，向数据表中添加数据，并且设置 PDOStatement 对象的 errorCode 属性，检测代码中的错误。关键代码如下：

```
<?php
if($_POST['Submit']=="提交" && $_POST['pdo']!=""){
	$dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
	$host='localhost';                          //数据库主机名
	$dbName='db_database16';                    //使用的数据库
	$user='root';                               //数据库连接用户名
	$pass='111';                                //对应的密码
	$dsn="$dbms:host=$host;dbname=$dbName";
	$pdo = new PDO($dsn, $user, $pass);         //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
	$query="insert into tb_pdo_mysqls(pdo_type,database_name,dates)values('".$_POST['pdo']."','".$_POST
['databases']."','".$_POST['dates']."')";
	$result=$pdo->prepare($query);
	$result->execute();
	$code=$result->errorCode();
	if(empty($code)){
		echo "数据添加成功！";
	}else{
		echo '数据库错误：<br/>';
		echo 'SQL Query:'.$query;
		echo  '<pre>';
		var_dump($result->errorInfo());
		echo '</pre>';
	}
}
?>
```

在本实例中，在定义 INSERT 添加语句时，使用了错误的数据表名称 tb_pdo_mysqls（正确名称是 tb_pdo_mysql），导致输出结果如图 16.5 所示。

图 16.5　在默认模式中捕获 SQL 中的错误

16.5.2　使用警告模式——PDO::ERRMODE_WARNING

警告模式会产生一个 PHP 警告，并设置 errorCode 属性。如果设置的是警告模式，那么除非明确地检查错误代码，否则程序将继续按照其方式运行。

例 16.05　设置警告模式，通过 prepare 和 execute 方法读取数据库中数据，并且通过 while 语句和 fetch()方法完成数据的循环输出，体会在设置成警告模式后执行错误的 SQL 语句，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.05）**

创建 index.php 文件，连接 MySQL 数据库，通过预处理语句 prepare()和 execute()执行 SELECT 查询语句，并设置一个错误的数据表名称，同时通过 setAttribute()方法设置为警告模式，最后通过 while 语句和 fetch()方法完成数据的循环输出。关键代码如下。

```
<?php
$dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
$host='localhost';                          //数据库主机名
$dbName='db_database16';                    //使用的数据库
$user='root';                               //数据库连接用户名
$pass='111';                                //对应的密码
$dsn="$dbms:host=$host;dbname=$dbName";
try {
    $pdo = new PDO($dsn, $user, $pass);     //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
     $pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_WARNING);   //设置为警告模式
     $query="select * from tb_pdo_mysqls";  //定义 SQL 语句
     $result=$pdo->prepare($query);         //准备查询语句
     $result->execute();                    //执行查询语句，并返回结果集
     while($res=$result->fetch(PDO::FETCH_ASSOC)){   //while 循环输出查询结果集，并且设置结果集为关联索引
     ?>
          <tr>
            <td height="22" align="center" valign="middle"><?php echo $res['id'];?></td>
            <td align="center" valign="middle"><?php echo $res['pdo_type'];?></td>
            <td align="center" valign="middle"><?php echo $res['database_name'];?></td>
            <td align="center" valign="middle"><?php echo $res['dates'];?></td>
          </tr>
<?php
     }
```

```
            } catch (PDOException $e) {
    die ("Error!: " . $e->getMessage() . "<br/>");
}
            ?>
```

在设置为警告模式后，如果 SQL 语句出现错误将给出一个提示信息，但是程序仍能够继续执行下去，运行结果如图 16.6 所示。



图 16.6　设置警告模式后捕获的 SQL 语句错误

16.5.3　使用异常模式——PDO::ERRMODE_EXCEPTION

异常模式会创建一个 PDOException，并设置 errorCode 属性。它可以将执行代码封装到一个 try…catch…语句块中。未捕获的异常将会导致脚本中断，并显示堆栈跟踪让用户了解是哪里出现的问题。

例 16.06　在执行数据库中数据的删除操作时，设置为异常模式，并且编写一个错误的 SQL 语句（操作错误的数据表 tb_pdo_mysqls），体会异常模式与警告模式和默认模式的区别，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.06）**

（1）创建 index.php 文件，连接 MySQL 数据库，通过预处理语句 prepare()和 execute()执行 SELECT 查询语句，通过 while 语句和 fetch()方法完成数据的循环输出，并且设置删除超链接，链接到 delete.php 文件，传递的参数是数据的 ID 值。运行结果如图 16.7 所示。



图 16.7　数据的循环输出

（2）创建 delete.php 文件，获取超链接传递的数据 ID 值，连接数据库，通过 setAttribute()方法设置为异常模式，定义 DELETE 删除语句，删除一个错误数据表（tb_pdo_mysqls）中的数据，并且通过 try{…}catch{…}语句捕获错误信息。代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );    //设置文件编码格式
if($_GET['conn_id']!=""){
     $dbms='mysql';//数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
     $host='localhost';                              //数据库主机名
```

```
        $dbName='db_database16';                    //使用的数据库
        $user='root';                               //数据库连接用户名
        $pass='111';                                //对应的密码
        $dsn="$dbms:host=$host;dbname=$dbName";
        try {
        $pdo = new PDO($dsn, $user, $pass);     //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
                $pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
                $query="delete from tb_pdo_mysqls where Id=:id";
                $result=$pdo->prepare($query);                  //预准备语句
                $result->bindParam(':id',$_GET['conn_id']);     //绑定更新的数据
                $result->execute();
        } catch (PDOException $e) {
                echo 'PDO Exception Caught.';
                echo 'Error with the database:<br/>';
                echo   'SQL Query: '.$query;
                echo '<pre>';
        echo "Error: " . $e->getMessage(). "<br/>";
                echo "Code: " . $e->getCode(). "<br/>";
                echo "File: " . $e->getFile(). "<br/>";
                echo "Line: " . $e->getLine(). "<br/>";
                echo "Trace: " . $e->getTraceAsString(). "<br/>";
                echo '</pre>';
        }
}
?>
```

在设置为异常模式后，执行错误的 SQL 语句返回的结果如图 16.8 所示。

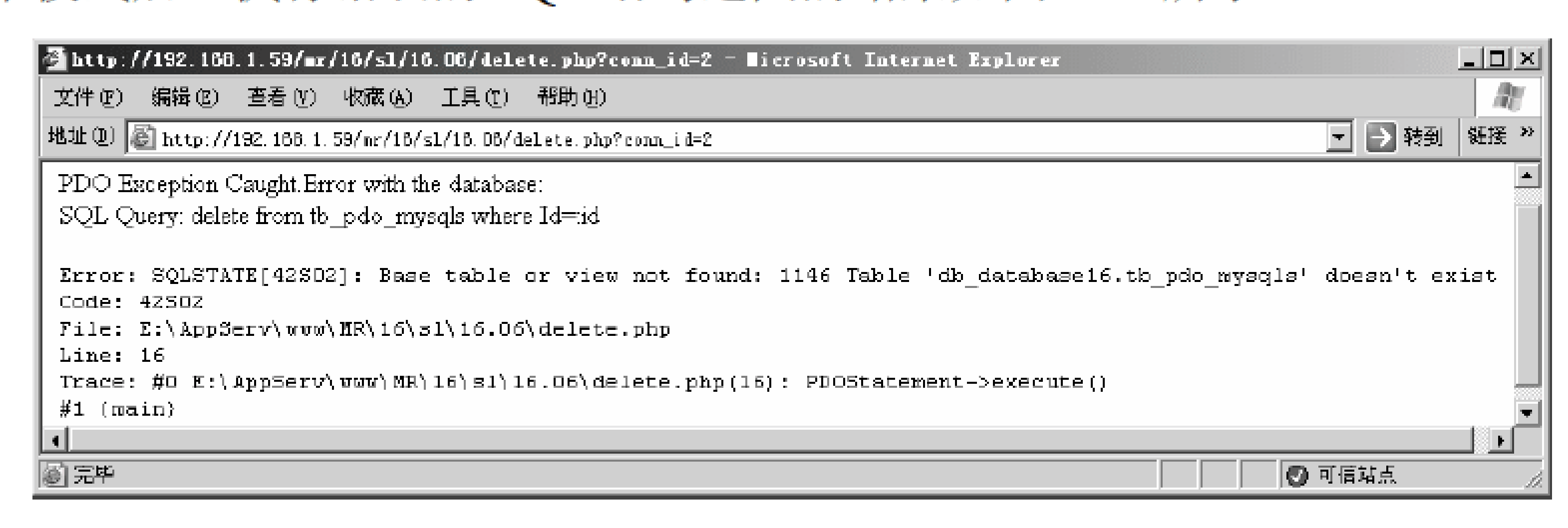


图 16.8　异常模式捕获的 SQL 语句错误信息

16.6　PDO 中错误处理

在 PDO 中有两个获取程序中错误信息的方法，即 errorCode()方法和 errorInfo()方法。

16.6.1　errorCode()方法

errorCode()方法用于获取在操作数据库句柄时所发生的错误代码，这些错误代码被称为 SQLSTATE 代码。其语法如下：

```
int PDOStatement::errorCode ( void )
```

errorCode()方法返回一个 SQLSTATE，SQLSTATE 是由 5 个数字和字母组成的代码。



例 16.07　在 PDO 中通过 query()方法完成数据的查询操作，并且通过 foreach 语句完成数据的循环输出。在定义 SQL 语句时使用一个错误的数据表，并且通过 errorCode()方法返回错误代码，具体步骤如下。（**实例位置：光盘\mr\16\sl\16.07**）

创建 index.php 文件。首先通过 PDO 连接 MySQL 数据库，然后通过 query()方法执行查询语句。接着通过 errorCode()方法获取错误代码，最后通过 foreach 语句完成数据的循环输出。关键代码如下：

```
    <?php
$dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
$host='localhost';                    //数据库主机名
$dbName='db_database16';              //使用的数据库
$user='root';                         //数据库连接用户名
$pass='111';                          //对应的密码
$dsn="$dbms:host=$host;dbname=$dbName";
try {
$pdo = new PDO($dsn, $user, $pass);   //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
$query="select * from tb_pdo_mysqls"; //定义 SQL 语句
$result=$pdo->query($query);          //执行查询语句，并返回结果集
echo "errorCode 为：".$pdo->errorCode();
foreach($result as $items){
    ?>
        <tr>
          <td height="22" align="center" valign="middle"><?php echo $items['id'];?></td>
          <td align="center" valign="middle"><?php echo $items['pdo_type'];?></td>
          <td align="center" valign="middle"><?php echo $items['database_name'];?></td>
          <td align="center" valign="middle"><?php echo $items['dates'];?></td>
        </tr>
         <?php
         }
    } catch (PDOException $e) {
    die ("Error!: " . $e->getMessage() . "<br/>");
}
         ?>
```

运行结果如图 16.9 所示。

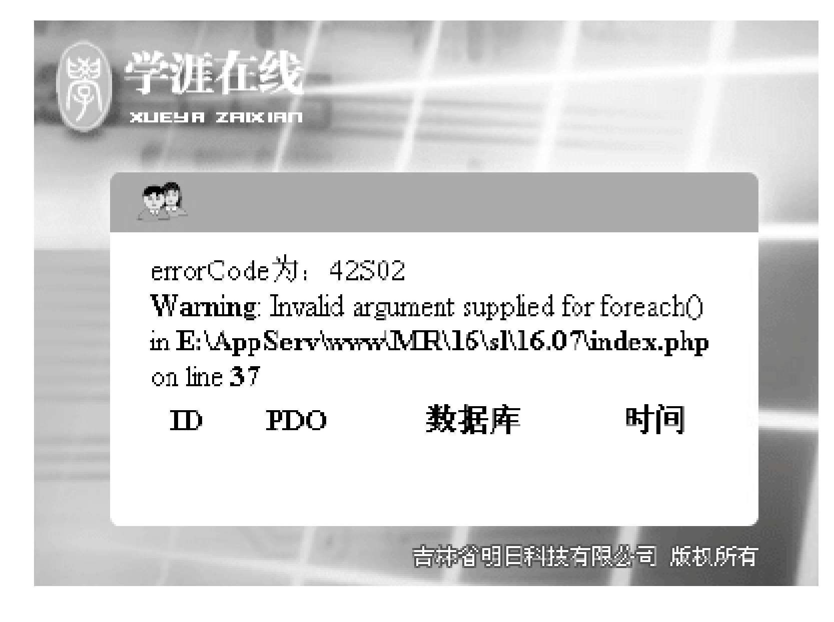


图 16.9　通过 errorCode()方法获取错误代码

16.6.2　errorInfo()方法

errorInfo()方法用于获取操作数据库句柄时所发生的错误信息。其语法如下：

```
array PDOStatement::errorInfo ( void )
```

errorInfo()方法的返回值为一个数组，它包含了相关的错误信息。

例 16.08 在 PDO 中通过 query()方法完成数据的查询操作，并且通过 foreach 语句完成数据的循环输出。在定义 SQL 语句时使用一个错误的数据表，并且通过 errorInfo()方法返回错误信息，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.08）**

创建 index.php 文件。首先通过 PDO 连接 MySQL 数据库，然后通过 query()方法执行查询语句，接着通过 errorInfo()方法获取错误信息，最后通过 foreach 语句完成数据的循环输出。关键代码如下：

```
<?php
$dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
$host='localhost';                          //数据库主机名
$dbName='db_database16';                    //使用的数据库
$user='root';                               //数据库连接用户名
$pass='111';                                //对应的密码
$dsn="$dbms:host=$host;dbname=$dbName";
try {
$pdo = new PDO($dsn, $user, $pass);                 //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
        $query="select * from tb_pdo_mysqls";       //定义 SQL 语句
        $result=$pdo->query($query);                //执行查询语句，并返回结果集
        print_r($pdo->errorInfo());
    foreach($result as $items){
    ?>
         <tr>
           <td height="22" align="center" valign="middle"><?php echo $items['id'];?></td>
           <td align="center" valign="middle"><?php echo $items['pdo_type'];?></td>
           <td align="center" valign="middle"><?php echo $items['database_name'];?></td>
           <td align="center" valign="middle"><?php echo $items['dates'];?></td>
         </tr>
          <?php
          }
          } catch (PDOException $e) {
    die ("Error!: " . $e->getMessage() . "<br/>");
}
          ?>
```

运行结果如图 16.10 所示。



图 16.10 通过 errorInfo()获取错误信息



16.7　PDO 中事务处理

在 PDO 中同样可以实现事务处理的功能，其应用的方法如下：

☑　开启事务——beginTransaction()方法

beginTransaction()方法将关闭自动提交（autocommit）模式，直到事务提交或者回滚以后才恢复。

☑　提交事务——commit()方法

commit()方法完成事务的提交操作，成功则返回 TRUE，否则返回 FALSE。

☑　事务回滚——rollBack()方法

rollBack()方法执行事务的回滚操作。

例 16.09　通过 prepare()和 execute()方法向数据库中添加数据，并且通过事务处理机制确保数据能够正确地添加到数据中，具体实现如下。**（实例位置：光盘\mr\16\sl\16.09）**

创建 index.php 文件。首先，定义数据库连接的参数，创建 try…catch…语句，在 try…语句中实例化 PDO 构造函数，完成与数据库的连接，并且通过 beginTransaction()方法开启事务。然后，定义 INSERT 添加语句，通过$_POST[]方法获取表单中提交的数据，通过 prepare()和 execute()方法向数据库中添加数据，并且通过 commit()方法完成事务的提交操作。最后，在 catch…语句中返回错误信息，并且通过 rollBack()执行事务的回滚操作。代码如下：

```
<?php
if($_POST['Submit']=="提交" && $_POST['pdo']!=""){
	$dbms='mysql';//数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
	$host='localhost';				//数据库主机名
	$dbName='db_database16';			//使用的数据库
	$user='root';					//数据库连接用户名
	$pass='111';					//对应的密码
	$dsn="$dbms:host=$host;dbname=$dbName";
	try {
	$pdo = new PDO($dsn, $user, $pass);	//初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
		$pdo->beginTransaction();		//开启事务
		$query="insert into tb_pdo_mysql(pdo_type,database_name,dates)values('".$_POST['pdo']."','".$_ POST
['databases']."','".$_POST['dates']."')";
		$result=$pdo->prepare($query);
		if($result->execute()){
			echo "数据添加成功！";
		}else{
			echo "数据添加失败！";
		}
		$pdo->commit();				//执行事务的提交操作
	} catch (PDOException $e) {
		die ("Error!: " . $e->getMessage() . "<br/>");
	$pdo->rollBack();				//执行事务的回滚
	}
}
?>
```

运行结果如图 16.11 所示。



图 16.11　数据添加中应用事务处理机制

16.8　PDO 中存储过程

存储过程允许在更接近于数据的位置操作数据，从而减少带宽的使用，它们使数据独立于脚本逻辑，允许使用不同语言的多个系统以相同的方式访问数据，从而节省了花费在编码和调试上的宝贵时间。同时它使用预定义的方案执行操作，提高了查询速度，并且能够阻止与数据的直接相互作用，从而起到保护数据的作用。

下面讲解如何在 PDO 中调用存储过程。这里首先创建一个存储过程，其 SQL 语句如下：

```
drop procedure if exists pro_reg;
delimiter //
create procedure pro_reg (in nc varchar(80), in pwd varchar(80), in email varchar(80),in address varchar(50))
begin
insert into tb_reg (name, pwd ,email ,address) values (nc, pwd, email, address);
end;
//
```

“drop”语句删除 MySQL 服务器中已经存在的存储过程 pro_reg。

“delimiter //”的作用是将语句结束符更改为“//”。

“in nc varchar(80)……in address varchar(50)”表示要向存储过程中传入的参数。

“begin……end”表示存储过程中的语句块，它的作用类似与 PHP 语言中的“{……}”。

存储过程创建成功后，下面调用这个存储过程实现用户注册的功能。

例 16.10　在 PDO 中通过 call 语句调用存储过程，实现用户注册信息的添加操作，具体步骤如下。**（实例位置：光盘\mr\16\sl\16.10）**

创建 index.php 文件。首先，创建 form 表单，将用户注册信息通过 POST 方法提交到本页。然后，在本页中编写 PHP 脚本，通过 PDO 连接 MySQL 数据库，并且设置数据库编码格式为 UTF-8，获取表单中提交的用户注册信息。接着，通过 call 语句调用存储过程 pro_reg，将用户注册信息添加到数据表中。最后，通过 try…catch…语句块返回错误信息。关键代码如下：

```
<?php
 if($_POST['submit']!=""){
 $dbms='mysql'; //数据库类型，对于开发者来说，使用不同的数据库，只需改这个即可，不用记住那么多的函数
     $host='localhost';                          //数据库主机名
     $dbName='db_database16';                    //使用的数据库
     $user='root';                               //数据库连接用户名
     $pass='111';                                //对应的密码
     $dsn="$dbms:host=$host;dbname=$dbName";
     try {
```

```
        $pdo = new PDO($dsn, $user, $pass);      //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
            $pdo->query("set names utf8");       //设置数据库编码格式
            $pdo->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION); //定义错误异常模式
        $nc=$_POST['nc'];
        $pwd=md5($_POST['pwd']);
        $email=$_POST['email'];
        $address=$_POST['address'];
            $query="call pro_reg('$nc','$pwd','$email','$address')";
            $result=$pdo->prepare($query);
            if($result->execute()){
                echo "数据添加成功！";
            }else{
                echo "数据添加失败！";
            }
        } catch (PDOException $e) {
        echo 'PDO Exception Caught.';
            echo 'Error with the database:<br/>';
            echo   'SQL Query: '.$query;
            echo '<pre>';
        echo "Error: " . $e->getMessage(). "<br/>";
            echo "Code: " . $e->getCode(). "<br/>";
            echo "File: " . $e->getFile(). "<br/>";
            echo "Line: " . $e->getLine(). "<br/>";
            echo "Trace: " . $e->getTraceAsString(). "<br/>";
            echo '</pre>';
        }
 }
?>
```

运行结果如图 16.12 所示。

图 16.12　通过存储过程完成用户的注册

16.9　照猫画虎——基本功训练

16.9.1　基本功训练 1——连接 MySQL 数据库

视频讲解：光盘\mr\16\lx\连接 MySQL 数据库.exe

实例位置：光盘\mr\16\zmhh\01

在学习通过 PDO 连接 MySQL 数据库之前，大家先创建一个 MySQL 数据库 db_database16，并且在

db_database16 数据库中创建数据表 tb_pdo，然后定义数据库连接的参数，最后，通过 PDO 构造函数创建连接。运行结果如图 16.13 所示。



图 16.13　连接 MySQL 数据库

操作步骤如下：

（1）创建脚本文件 index.php。编写连接 MySQL 数据库的连接参数，其中包括$dbms、$dbName、$user、$pwd 和$host。

（2）实例化 PDO 对象，输出连接成功的提示并利用 try…catch…语句捕获异常。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");			//设置页面的编码格式
	$dbms='mysql';								//数据库类型
	$dbName='db_database16';						//使用的数据库名称
	$user='root';								//使用的数据库用户名
	$pwd='111';								//使用的数据库密码
	$host='localhost';							//使用的主机名称
	$dsn="$dbms:host=$host;dbname=$dbName";
	try {									//捕获异常
		$pdo=new PDO($dsn,$user,$pwd);				//实例化对象
		echo "PDO 连接 MySQL 成功";
	} catch (Exception $e) {
		echo $e->getMessage()."<br>";
	}
?>
```

照猫画虎：在上面实例中，将$dbName 属性值设置为 db_database016，然后查看显示结果，体会异常提示。(20 分)(实例位置：光盘\mr\16\zmhh\01_zmhh)

16.9.2　基本功训练 2——通过 PDO 向数据库中添加数据

视频讲解：光盘\mr\16\lx\通过 PDO 向数据库中添加数据.exe

实例位置：光盘\mr\16\zmhh\02

PDO 数据库抽象层的主要特点是为不同的数据库提供统一的接口，使用户在后期维护或数据库变更时减少麻烦等。本实例通过 PDO 向已经创建好的数据库中添加数据，运行结果如图 16.14 所示。



图 16.14　向数据库中添加数据

操作步骤如下：

（1）创建脚本文件 index.php，在此脚本文件中编写表单，其中包括文本框、密码框、提交按钮和重置按钮。

（2）当单击“确定”按钮时，利用$_POST[ ]全局数组获取文本框和密码框的信息，并将其作为 SQL 语句的参数。SQL 语句的代码如下：

```
$sql="insert into tb_pdo values('','$_POST[text]','$_POST[pwd]',now())";
```

（3）定义 PDO 抽象层的参数，其中包括$dbms、$user、$pwd、$host 等，拼接$dsn。

（4）实例化 PDO 对象，利用对象句柄调用 exec()函数向数据库中插入数据信息，并显示插入数据的条数。核心代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");				//设置编码集
```

```
    $dbms='mysql';                                                    //定义 PDO 相关参数
    $dbName='db_database16';
    $user='root';
    $pwd='111';
    $host='localhost';
    $dsn="$dbms:host=$host;dbname=$dbName";
    if(isset($_POST[sub])){                                           //当页面中存在 sub 变量时
        if($_POST[text]==""&&$_POST[pwd]==""){                        //当文本框为空时输出提示
            echo "文本框内容不能为空";
        }else{
            try {                                                     //try…catch…捕捉异常
                $pdo=new PDO($dsn,$user,$pwd);                        //实例化对象
                $sql="insert into tb_pdo values('','$_POST[text]','$_POST[pwd]',now())";//SQL 语句
                $result=$pdo->exec($sql);
                echo "插入数据成功，影响条数为".$result;                  //返回结果集
            } catch (Exception $e) {
                echo "ERROR!!".$e->getMessage()."<br>";               //输出异常
            }
        }
    }
?>
```

照猫画虎：扩展上面实例的相关知识，利用错误处理模式捕获程序中的错误。（20 分）（实例位置：光盘\mr\16\zmhh\02_zmhh）

16.9.3　基本功训练 3——通过 PDO 浏览数据库中数据

视频讲解：光盘\mr\16\lx\通过 PDO 浏览数据库中数据.exe

实例位置：光盘\mr\16\zmhh\03

利用 PDO 抽象层显示数据需要使用函数 query()，此函数与普通的操作数据库函数并无区别，只不过在 PDO 中此函数只用于数据的查询，而插入、修改和更新的方法需要使用 exec()函数来实现。运行结果如图 16.15 所示。

ID	用户名称	用户密码	操作时间
1	mr	mrsoft	2010-11-24
2	mrsoft	mrkj	2010-11-24
3	ym	066066	2010-11-24
4	pkh	123456	2010-11-24

图 16.15　浏览数据库中的数据

操作步骤如下：

首先，创建脚本文件 index.php，编写表格，设置页面的编码。其次，定义 PDO 所需的相关参数，拼接 dsn 参数，利用对象句柄$pdo 调用 PDO 抽象层的 query()方法并返回结果集。最后，将结果集通过 foreach()语句循环输出到表格中。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                  //定义页面编码风格
    $dbms='mysql';                                                    //定义 PDO 的相关参数
    $dbName='db_database16';
    $user='root';
    $pwd='111';
    $host='localhost';
    $dsn="$dbms:host=$host;dbname=$dbName";
?>
```

```
<table>                                                              //表格输出数据内容
<tr>
        <td class="one">ID</td>
        <td class="one">用户名称</td>
        <td class="one">用户密码</td>
        <td class="one">操作时间</td>
</tr>
<?php
    try {                                                            //利用 try…catch…语句捕获异常
        $pdo=new PDO($dsn,$user,$pwd);
        $sql="select * from tb_pdo";
        $result=$pdo->query($sql);
        foreach($result as $key=>$value){
?>
    <tr>
        <td><?php echo $value[0];?></td>                             //循环输出数据并定义到表格中
        <td><?php echo $value[1];?></td>
        <td><?php echo $value[2];?></td>
        <td><?php echo $value[3];?></td>
    </tr>
<?php
        }
    } catch (Exception $e) {
        echo "ERROR!!".$e->getMessage()."<br>";
    }
?>
```

照猫画虎：应用 PDO 的知识，使用警告模式获取 SQL 语句错误并使程序依照上例实现数据的输出。（20 分）（实例位置：光盘\mr\16\zmhh\03_zmhh）

16.9.4　基本功训练 4——通过 PDO 更新数据库中数据

视频讲解：光盘\mr\16\lx\通过 PDO 更新数据库中数据.exe

实例位置：光盘\mr\16\zmhh\04

在利用 MySQL 数据库函数更新数据时，需要将更改的数据信息 id 拼接到地址栏的参数中。在 PDO 中操作数据更新也不例外。虽然利用 PDO 操作数据的方法是不同的，但原理是相同的。本实例运行结果如图 16.16 和图 16.17 所示。

ID	用户名称	用户密码	操作时间	修改
1	mr	mrsoft	2010-11-24	修改
2	mrsoft	mrkj	2010-11-24	修改
3	ym	066066	2010-11-24	修改
4	pkh	123456	2010-11-24	修改
3	yangming	••••••	2010-11-24	确定

图 16.16　更新数据



图 16.17　更新数据成功

操作步骤如下：

首先，创建脚本文件 index.php。编写表格，设置 PDO 抽象层的相关参数拼接$dsn 变量。其次，判断“修改”按钮是否被单击，当“修改”按钮被单击时，获取地址栏传递的参数并在表格最下方动态创建<form>



表单，其中包括 3 个文本框和一个提交按钮。文本框中的 value 属性值由 PHP 提供数据并设置 id 的文本框，使其设置为只读属性不可更改。最后，当单击“确定”按钮时，通过 PDO 的 exec()方法执行更新操作。代码如下：

```
<?php
	try {
		$pdo=new PDO($dsn,$user,$pwd);                                  //实例化对象
		$sql="select * from tb_pdo";                                     //查询 SQL 语句
		$result=$pdo->query($sql);                                       //返回结果集
		foreach($result as $value){
?>
	<tr>
		<td class="one"><?php echo $value[0];?></td>                    //循环输出数据
		<td class="one"><?php echo $value[1];?></td>
		<td class="one"><?php echo $value[2];?></td>
		<td class="one"><?php echo $value[3];?></td>
		<td class="one"><a href="index.php?id=<?php echo $value[0];?>">修改</a></td>      //拼接地址栏参数
	</tr>
<form action="method="post">
<?php
		}
		if(isset($_GET[id])){                                            //当页面中存在 id 变量时
			$sql1="select * from tb_pdo where id='$_GET[id]'";          //拼接 SQL 语句
			$resul=$pdo->query($sql1);                                   //返回结果集
				foreach ($resul as $value){                              //循环输出结果


?>
	<tr>
			<td class="one"><input readonly type='text' name='id' value='<?php echo $value[id];?>'></td>
			<td class="one"><input type='text' name='user' value='<?php echo $value[1];?>'></td>
			<td class="one"><input type='password' name='pwd' value='<?php echo $value[2];?>'></td>
			<td class="one"><input type='text' name='date' value='<?php echo $value[3];?>'></td>
			<td class="one"><input class="two" type='submit' name='sub1' value='确定'></td>
	</tr>

<?php
			}
		}
?>
	</form>
<?php
		if($_POST[sub1]=="确定"){                                         //当页面中存在参数 sub1 时
			$sql2="update tb_pdo set username='$_POST[user]',userpwd='$_POST[pwd]',date='$_POST
[date]' where id='$_POST[id]'";                                          //拼接 SQL 语句
			$resu=$pdo->exec($sql2);                                     //执行更新操作
			if($resu==1){
				echo "<script>alert('更新数据成功');window.location.href='index.php'</script>";
			}
		}
	} catch (Exception $e) {
```

```
            echo "ERROR!!!".$e->getMessage()."<br>";
        }
?>
```

照猫画虎：在上面实例中，用户根据具体情况，使用预处理语句执行数据更新操作。（20 分）（实例位置：光盘\mr\16\zmhh\04_zmhh）

16.9.5　基本功训练 5——连接 MS SQL Server 数据库

视频讲解：光盘\mr\16\lx\连接 MS SQL Server 数据库.exe

实例位置：光盘\mr\16\zmhh\05

PDO 这种抽象层的概念可以实现与主流的数据库使用同一接口进行连接。连接的方法是定义指定的 PDO 参数，然后利用同一方法操作即可。本实例通过 PDO 连接 MS SQL Server 数据库，运行结果如图 16.18 所示。



图 16.18　连接 MS SQL Server 数据库

实现本实例的操作非常简单，只需更改 PDO 参数$dbms 为 mssql，$user 为 sa，$pwd 为""（空）即可。然后使用与连接 MySQL 数据库相同的方法实现连接 MS SQL Server 数据库。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");          //设置页面的编码风格
    $host='PC-201006101638';                                 //设置主机名称
    $user='sa';                                              //设置用户名
    $pwd='';                                                 //设置密码
    $dbName='db_database16';                                 //设置所要连接的数据库
    $dbms='mssql';
    $dsn="mssql:host=$host;dbname=$dbName";
    try {                                                    //利用 try…catch…捕捉异常并实现连接
        $pdo = new PDO($dsn,$user,$pwd);
        echo "连接 MS SQL Server 成功";
    } catch (Exception $e) {
        die("ERROR!".$e->getMessage()."<br>");
    }
?>
```

照猫画虎：对上面的实例进行扩展，实现向 MS SQL Server 中添加数据。（20 分）（实例位置：光盘\mr\16\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							



16.10　情景应用——拓展与实践

16.10.1　情景应用 1——明日书店会员注册

视频讲解：光盘\mr\16\lx\明日书店会员注册.exe

实例位置：光盘\mr\16\qjyy\01

注册系统在本书的很多章节中都有体现，本次所讲解的是利用 PDO 抽象层知识实现的会员注册系统。其核心思想是拼接插入的 SQL 代码，将文本框的相关信息动态地添加到数据表中。运行结果如图 16.19 所示。



图 16.19　会员注册系统

操作步骤如下：

首先，创建脚本文件 index.php。在脚本文件中编写表单，设置一个提交按钮和一个重置按钮。其次，当单击“注册”按钮时，首先通过 PDO 连接 MySQL 数据库，实例化 PDO 对象并利用对象句柄调用 exec()方法，向数据库中添加数据。代码如下：

```
<?php
    $dbms='mysql';                                        //定义 PDO 的相关参数
    $host='localhost';
    $user='root';
    $pwd='111';
    $dbName='db_database16';
    $dsn="$dbms:host=$host;dbname=$dbName";
    if(isset($_POST[sub])){                               //当页面中存在 sub 变量时
        try {
            $pdo=new PDO($dsn,$user,$pwd);                //实例化对象
            $sql="insert into tb_zc values('','$_POST[text]','$_POST[pwd]','$_POST[qq]','$_POST[mail]',now())";
                                                          //拼接 SQL 语句
            $result=$pdo->exec($sql);                     //返回结果集
            if($result==1){                               //输出提示
                echo "<script>alert('注册成功');</script>";
```

```
            }
        } catch (Exception $e) {
            echo "ERROR".$e->getMessage()."<br>";        //输出异常
        }
    }
?>
```

DIY：根据上述内容，利用 PDO 中的预处理语句将文本框中的信息保存在 MS SQL Server 数据库中的 tb_pdo 表中。（20 分）（实例位置：光盘\mr\16\qjyy\01_diy）

16.10.2 情景应用 2——明日书店会员登录

视频讲解：光盘\mr\16\lx\明日书店会员登录.exe

实例位置：光盘\mr\16\qjyy\02

本例中的会员登录系统与 16.10.1 小节的实现原理基本相同，但是为了让用户对本小节内容有深入的了解，特在此实例中添加了验证操作，即当用户输入昵称为 mr，密码为 mrsoft 时才会显示登录成功。运行结果如图 16.20 所示。



图 16.20　会员登录

操作步骤如下：

首先，创建脚本文件，编写会员登录的 form 表单。其次，当单击“登录”按钮时，通过$_POST[]全局数组获取文本框提交的数据，并与数据库中的数据进行比较，当匹配成功时显示登录成功，否则不显示任何提示信息。核心代码如下：

```
<?php
    $dbms="mysql";                                          //定义 PDO 的相关参数
    $user="root";
    $pwd="111";
    $host="localhost";
    $dbName="db_database16";
    $dsn="$dbms:host=$host;dbname=$dbName";
    if(isset($_POST[sub])){                                 //当页面中存在 sub 变量时
        try {
            $pdo = new PDO($dsn,$user,$pwd);                //实例化对象
            $sql="select * from tb_pdo";                    //定义查询 SQL 语句
            $rs=$pdo->query($sql);                          //执行查询操作并返回结果集
            $array=array();                                 //定义空数组
            $arr=array();
            foreach($rs as $value){                         //定义循环语句并将取得的数据值压入数组
                array_push($array,$value[1]);
                array_push($arr,$value[2]);
            }
            foreach($array as $val){
                foreach($arr as $valu){
                    if($_POST[text]==$val and $_POST[pwd]==$valu){    //当匹配成功时输出提示
                        echo "<b>登录成功</b>";
                        exit;
                    }
                }
            }
```

```
            } catch (Exception $e) {
                echo "ERROR".$e->getMessage()."<br>";
            }
        }
?>
```

DIY：将 MySQL 和 MS SQL Server 数据库中的数据实现相互导入导出的操作。（20 分）（实例位置：光盘\mr\16\qjyy\02_diy）

16.10.3　情景应用 3——添加留言信息

视频讲解：光盘\mr\16\lx\添加留言信息.exe

实例位置：光盘\mr\16\qjyy\03

添加留言信息是非常普通和常见的模块操作，其实从原理上说与注册登录操作几乎一样，笔者意在告知读者 PDO 操作常见的几种模式，使读者熟能生巧。运行结果如图 16.21 所示。



图 16.21　添加留言信息

本例与注册登录模块原理相同，这里就不再阐述操作步骤了。核心代码如下：

```
<?php
    $dbms='mysql';                                          //定义 PDO 的相关参数
    $user='root';
    $pwd='111';
    $host='localhost';
    $dbName='db_database16';
    $dsn="$dbms:host=$host;dbname=$dbName";
    if(isset($_POST[sub])){                                 //当页面中存在 sub 变量时
        try {                                               //利用 try…catch…捕获异常
            $pdo = new PDO($dsn,$user,$pwd);                //实例化对象
            $pdo->query("SET NAMES utf8");                  //设置页面的编码集风格
            $sql="insert into tb_fb values('','$_POST[title]','$_POST[content]',now())";  //定义 SQL 语句
            $rs=$pdo->exec($sql);                           //执行插入操作
            if($rs=="1"){
                echo "<b>新闻发布成功</b>";
            }
        } catch (Exception $e) {
            echo "ERROR".$e->getMessage()."<br>";
        }
    }
?>
```

DIY：添加 errorInfo()方法实现获取操作数据库句柄时所发生的错误信息并实现以上应用的功能。（20 分）（实例位置：光盘\mr\16\qjyy\03_diy）

16.10.4　情景应用 4——浏览客户留言

视频讲解：光盘\mr\16\lx\浏览客户留言.exe

实例位置：光盘\mr\16\qjyy\04

本例是基本功训练 3 的具体应用，唯一不同的是笔者在本例中并没有采用分页输出数据，只是利用 SQL 语句的排序方法，根据 ID 进行降幂排列而已。运行结果如图 16.22 所示。

浏览客户留言

ID	标题	内容	日期
9	你好编程宝典	编程宝典是您最好的选择	2010-11-24
8	你好编程宝典	编程宝典是您最好的选择	2010-11-24
7	你好编程宝典	编程宝典是您最好的选择	2010-11-24
6	你好编程宝典	编程宝典是您最好的选择	2010-11-24

图 16.22　浏览客户留言

操作步骤如下：

（1）创建脚本文件 index.php。定义 PDO 抽象层的相关参数，其中包括$host、$user、$pwd、$dbName 和$dbms 等。

（2）在 try…catch…语句内部实例化 PDO 对象，利用对象句柄调用 query()方法并返回结果集，最后将返回的结果输出到表格中。代码如下：

```
<?php
	$dbms="mysql";                                          //定义 PDO 的相关参数
	$user="root";
	$pwd="111";
	$host="localhost";
	$dbName="db_database16";
	$dsn="$dbms:host=$host;dbname=$dbName";
	try {                                                   //利用 try…catch…捕获异常
		$pdo = new PDO($dsn,$user,$pwd);                    //实例化 PDO 对象
		$pdo->query("SET NAMES utf8");                      //设置页面数据的编码风格
		$sql="select * from tb_fb order by id desc limit 4";  //定义 SQL 语句
		$rs=$pdo->query($sql);                              //执行查询操作
		foreach($rs as $value){
?>
	<tr>
		<td class="two"><?php echo $value[0];?></td>        //循环输出数据
		<td class="two"><?php echo $value[1];?></td>
		<td class="two"><?php echo $value[2];?></td>
		<td class="two"><?php echo $value[3];?></td>
	</tr>
<?php
		}
	} catch (Exception $e) {
```

```
        echo "ERROR!!!".$e->getMessage()."<br>";
    }
?>
```

DIY：在上面实例中，利用 fetch()函数输出数据表数据。(20 分)(实例位置：光盘\mr\16\qjyy\04_diy)

16.10.5　情景应用 5——查询留言内容

视频讲解：光盘\mr\16\lx\查询留言内容.exe

实例位置：光盘\mr\16\qjyy\05

本例中的查询留言内容模块是一个典型的站内搜索关键字描红的基础操作。究其根源，其实 PDO 的操作并不复杂，它只是提供了一个连接多种数据库的一个统一的接口。至于具体的操作还是来自于 SQL 语句本身。运行结果如图 16.23 所示。

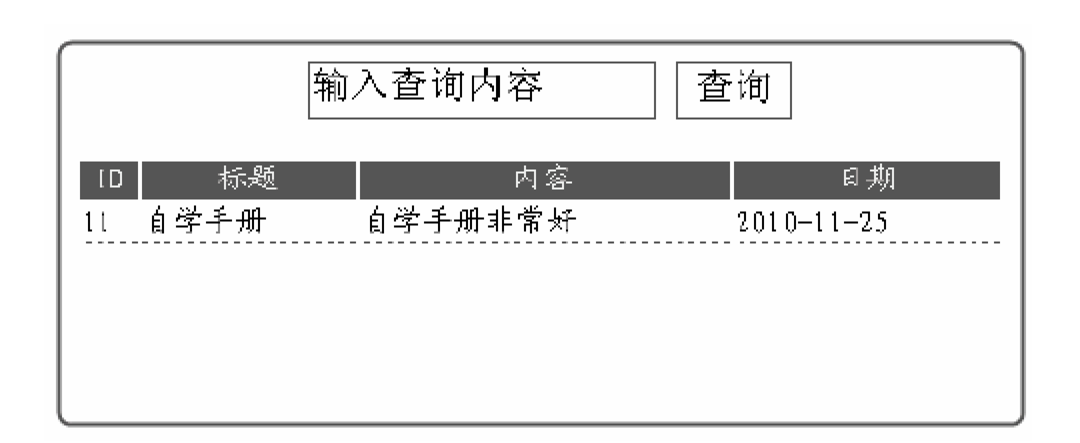


图 16.23　查询留言内容

操作步骤如下：

（1）创建脚本文件 index.php。定义<form>表单，设置一个文本框和一个“查询”按钮。

（2）当“查询”按钮被单击时，首先，判断文本框内容是否为空。其次，使用 PDO 抽象层连接 MySQL 数据库，并在 try…catch…内部利用 PDO 对象句柄调用 query()函数执行查询操作。最后，输出查询结果。核心代码如下：

```
<?php
    if(isset($_POST[sub])){                                          //判断页面是否存在 sub 变量
        if($_POST[text]=="" || $_POST[text]=="输入查询内容"){          //判断文本框内容是否为空
            echo "文本框内容不能为空";
        }else{
            $dbms="mysql";                                           //定义 PDO 的相关参数
            $user="root";
            $pwd="111";
            $host="localhost";
            $dbName="db_database16";
            $dsn="$dbms:host=$host;dbname=$dbName";
            try {                                                    //try…catch…捕获异常
                $pdo = new PDO($dsn,$user,$pwd);                     //实例化对象
                $sql="select * from tb_fb where title like '%".$_POST['text']."%'";     //拼接 SQL 语句
                $pdo->query("SET NAMES utf8");                       //设置页面数据的编码风格
                $rs=$pdo->query($sql);
                foreach($rs as $value){                              //将数据循环输出
    ?>
        <tr>
            <td class="f"><?php echo $rs [0];?></td>
```

```
                        <td class="f"><?php echo $rs [1];?></td>
                        <td class="f"><?php echo $rs [2];?></td>
                        <td class="f"><?php echo $rs [3];?></td>
                </tr>
<?php
                                }
                        } catch (Exception $e) {
                                echo "ERROR".$e->getMessage()."<br>";
                        }
                }
        }
?>
```

DIY：同样实现上面的实例，用户根据自己的需要将查询关键字描红输出。（20 分）（实例位置：光盘\mr\16\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

16.11 自我测试

一、选择题（每题 10 分，5 道题）

1．下列有关 PDO 的索引是关联索引的函数是（　　）。

A．PDO::LOWER　　B．PDO::ASSOC　　C．PDO::FETCH_NUM

2．如果在 PHP 中使用 Oracle 数据库作为数据库服务器，应该在 PDO 中加载（　　）驱动程序。

A．PDO_DBLIB　　B．PDO_MYSQL　　C．PDO_OCI　　D．PDO_ODBC

3．当 PDO 对象创建成功以后，与数据库的连接已经建立，就可以使用 PDO 对象了，下面（　　）不是 PDO 对象中的成员方法。

A．phpInfo()　　B．query()　　C．exec()　　D．prepare()

4．PDO 提供了多种不同的错误处理模式，不仅可以满足不同风格的编程，也可以调整扩展处理错误的方式。下面（　　）不是 PDO 提供的错误处理模式。

A．ERRMODE_SILENT　　B．ERRMODE_WARNING

C．PDO::ERRMODE_ERROR　　D．ERRMODE_EXCEPTION

5．下面（　　）函数属于 PDO 预处理语句。

A．PDO::ATTR_AUTOCOMMIT　　B．PREPARE

C．PDO::ERRMODE_ERROR　　D．ERRMODE_EXCEPTION

二、填空题（每题 10 分，5 道题）

1．开启 PDO 错误默认模式代码为（　　　　　　　　　　　　）。

2．开启 PDO 错误警告模式代码为（　　　　　　　　　　　　）。

3．开启 PDO 错误异常模式代码为（　　　　　　　　　　　　　　）。

4．执行 PDO 插入操作需要使用函数（　　　　　）。

5．假设 MySQL 数据库名称为 db_database16，数据库用户名为 root，密码为 111，数据源为 localhost，程序运行后页面中显示的内容为（　　）。

```
header("Content-Type:text/html;charset=utf-8");
$dbms='mysql';
$dbName='db_database16';
$user='root';
$pwd='111';
$host='localhost';
$dsn="$dbms:host=$host;dbname=$dbName";
try {
    $pdo=new PDO($dsn,$user,$pwd);
    echo "PDO 连接 MySQL 成功";
} catch (Exception $e) {
    echo $e->getMessage()."<br>";
}
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

16.12　行 动 指 南

开始日期：______年____月____日　　　　　　　　　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）利用面向对象的知识封装两个数据库连接类，其中一个类使用 PDO 操作，另外一个使用 MySQL 函数操作，然后通过接口传入对象不同实现数据库之间的切换。	
		（2）利用 PDO 的相关知识编写留言板。	
		（3）利用 PDO 的相关知识编写简单的新闻发布系统。	
		（4）利用 PDO 的事务处理机制编写简单的 OA 系统。	

续表

序号	内　　容	行 动 指 南
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

16.13　成功可以复制——中国第一程序员求伯君

成就：国内著名程序员，国内第一套文字处理软件 WPS 开发者。200 年荣获“中国 IT 十大风云人物”称号。

创始公司：创立金山软件股份有限公司，2007 年在香港主板挂牌上市。

财富巅峰：个人资产近 10 亿

出生日期：1964 年 11 月 26 日

最高学历：本科

超级语录：如果从开始就想着怎样赚钱，我也不会有今天。事业和金钱无关。当你全身心投入开发的时候，不给你钱你也要干。开发时，根本没有心思考虑报酬。只有先成就了业，才有资格谈报酬。

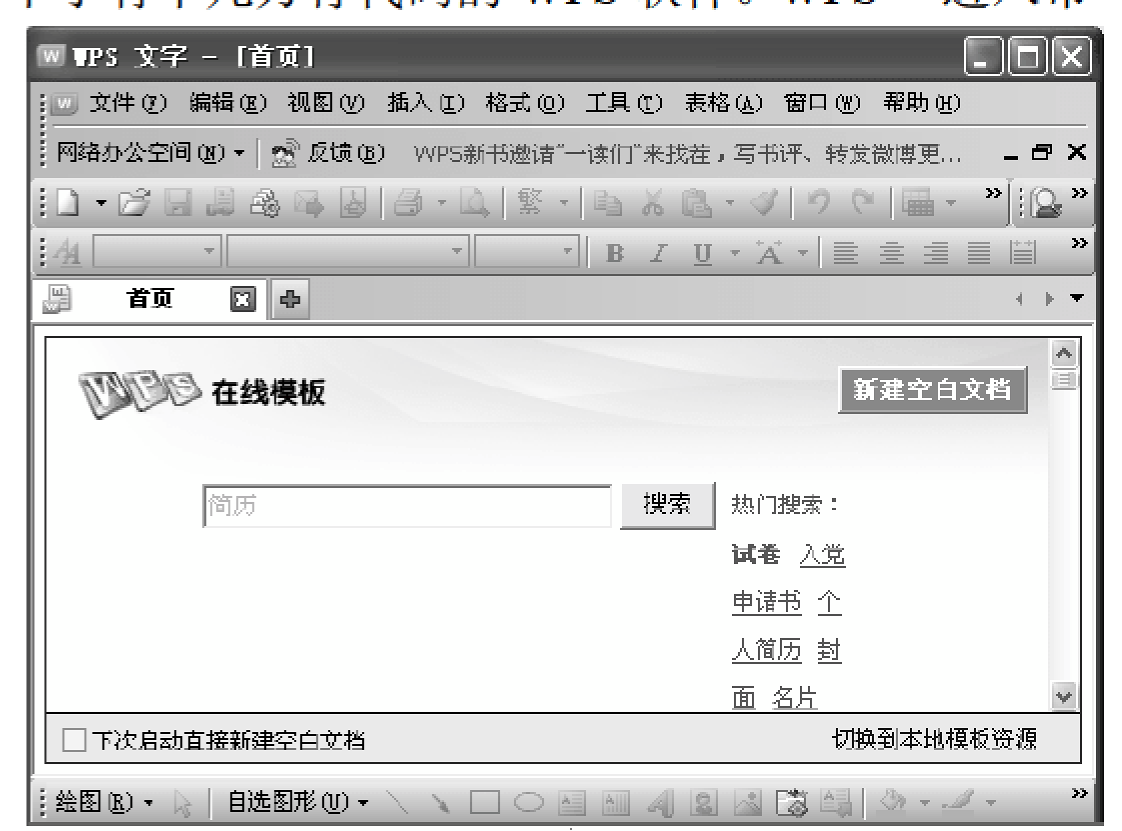
成功历程

求伯君，1964 年 11 月 26 日出生于浙江新昌县。1980 年，求伯君考入国防科技大学系统工程与数学系学习，因系统工程需要大量使用电脑，求伯君也比同龄人有了更早接触电脑的机会，这也成为求伯君的第一次人生机遇。1983 年，求伯君利用业余时间为国防科技大学图书馆开发了图书馆借还书管理系统，这是求伯君写的第一个能用的程序，为此求伯君获得了 40 多元钱的奖励，从此，求伯君与程序结下了不解之缘。

1986 年，求伯君改进编写了一个支持多种打印机的西山文字打印驱动程序，以 2000 元的价格卖给了四通，求伯君也成为四通的一员。西山文字打印驱动程序，四通定价 500 百元一套，卖了好几百套。

1988 年，求伯君离开四通公司，加入香港金山公司，负责开发用于 DOS 系统的中文文字处理软件——WPS。为了加快开发速度，求伯君把自己关在深圳的一个房间里，只要是醒着，就不停地写。什么

时候困了，就睡一会儿，饿了就吃方便面。在这样的一年零四个月中，求伯君生了三次病，第一次肝炎，第二次肝炎复发，第三次又复发，每次住院一个月到两个月。第二次肝炎复发正是软件开发最紧要的关头，求伯君把电脑搬到病房里继续写。求伯君在这孤独中，写下了有十几万行代码的 WPS 软件。WPS 一进入市场就凭着良好的口碑迅速占领市场，当年卖了 3 万多套，销售额达 6000 多万元。



WPS 软件界面

1994 年，求伯君在珠海成立珠海金山电脑公司，继续从事办公软件开发。开发的第一个软件产品——盘古组件（里面有 WPS、电子表和字典）遇到了 Word 的挑战，最终以失败告终。1996 年金山陷入苦难时期，运营资金紧张、员工信心不足。在危难时刻，求伯君毅然卖掉了别墅，凭着对 WPS 的深厚感情克服了重重困难与障碍，1997 年推出 WPS 97。WPS 97 推出仅两个多月，就销出了 13000 套，金山也借此迈过最艰难的时刻。

如今的金山软件，已占据通用软件全国第一，2009 年总营业收入达 10.22 亿元。

深度评价

求伯君的成功告诉我们，热爱编程的程序员们要耐得住寂寞，坚定信念，用自己的热情全身心地投入才能创造出成功的事业。

第17堂课

Smarty 模板技术

（视频讲解：162 分钟）

目前网络上针对 PHP 的模板数不胜数。作为最早的 MVC 模板之一，Smarty 在功能和速度上处于绝对的领先。那么，Smarty 的特点是什么？它是如何完成代码分离的？学习了本堂课知识用户会对此有深刻了解。

学习摘要：

- **走进 Smarty 模板引擎**
- **Smarty 模板设计**
- **Smarty 程序设计**

17.1 Smarty 模板引擎

Smarty 是一个使用 PHP 编写的 PHP 模板引擎，是目前业界最著名、功能最强大的一种 PHP 模板引擎。它将一个应用程序分成视图和逻辑控制两部分，也就是将 UI（用户界面）和 PHP code（PHP 代码）分离。这样，程序员在修改程序时不会影响到页面设计，而美工在重新设计或修改页面时也不会影响程序逻辑。Smarty 模板引擎的运行流程如图 17.1 所示。

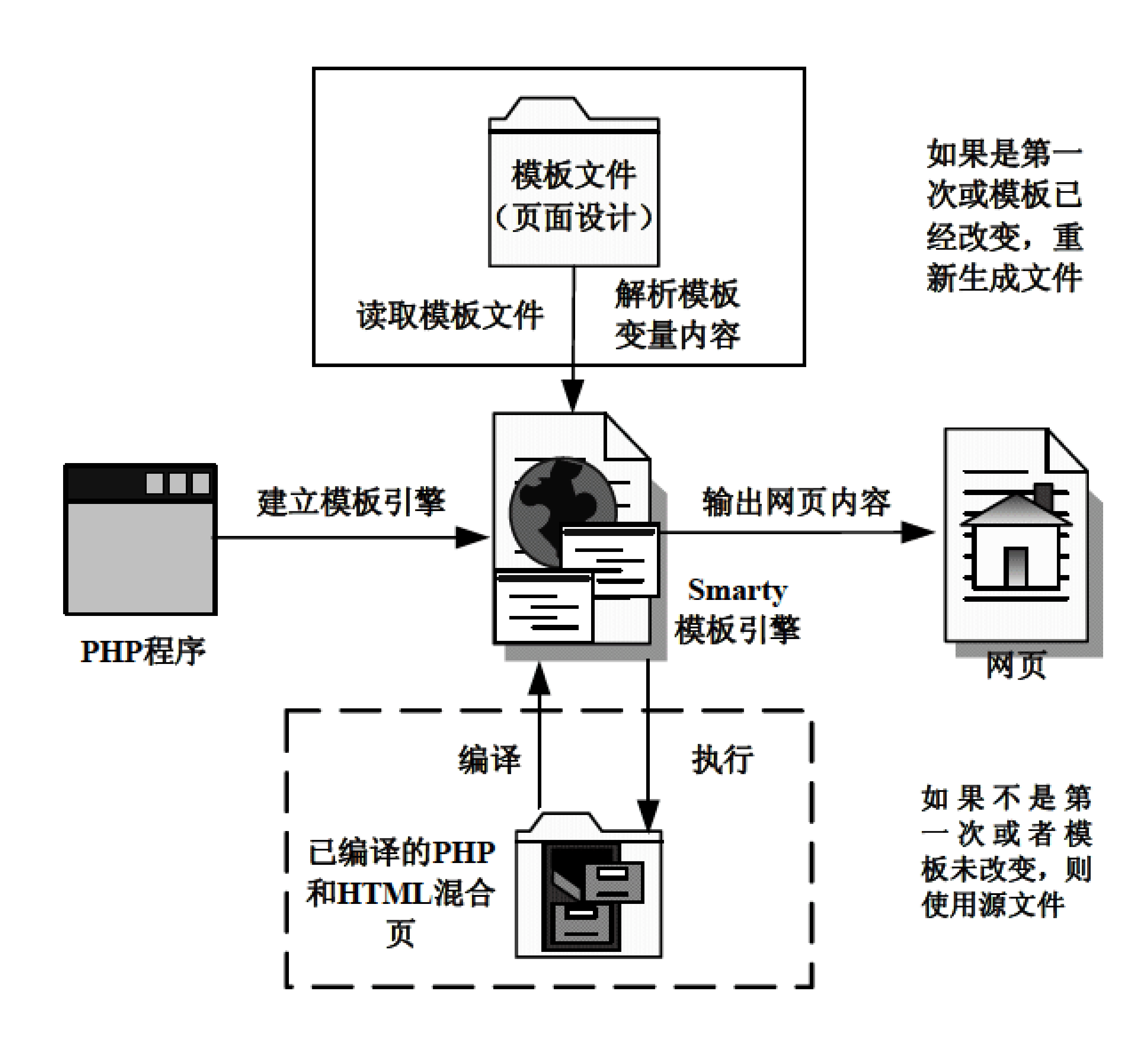


图 17.1 Smarty 模板引擎的运行流程

Smarty 拥有丰富的函数库，从统计字数到字符串的截取、文字的环绕以及正则表达式都可以直接使用，还具有很强的扩展能力。Smarty 模板的优点总结如下。

- ☑ 速度：相对于其他模板而言，采用 Smarty 模板编写的程序可以获得最快的速度。
- ☑ 编译性调用：采用 Smarty 模板编写的程序在运行时会生成一个 PHP 和 HTML 混合的文件，在下一次访问模板时会直接访问这个混编的文件（前提是源文件没有改变），而不必重新编译，进而可以提高访问速度。
- ☑ 缓存技术：Smarty 提供一种可选择的缓存技术，可以将客户端的 HTML 文件缓存成一个静态页。当用户开启缓存后，在指定的时间内，Web 请求会直接调用这个缓存文件，即直接调用静态的 HTML 文件。
- ☑ 插件技术：因为 Smarty 模板引擎是通过 PHP 面向对象技术实现的，所以不仅可以修改 Smarty 模板的源文件，而且可以通过自定义函数向 Smarty 中添加功能。
- ☑ 模板中可以使用 if/elseif/else/endif。

17.1.1 Smarty 模板引擎下载

PHP 没有内置 Smarty 模板类，需要单独下载和配置。并且 Smarty 要求服务器上的 PHP 版本最低为 4.0.6。



用户可以在 http://smarty.net/download.php 网站上下载最新的 Smarty 压缩包。本书中使用的版本是 Smarty-2.6.23。下载地址如图 17.2 所示。

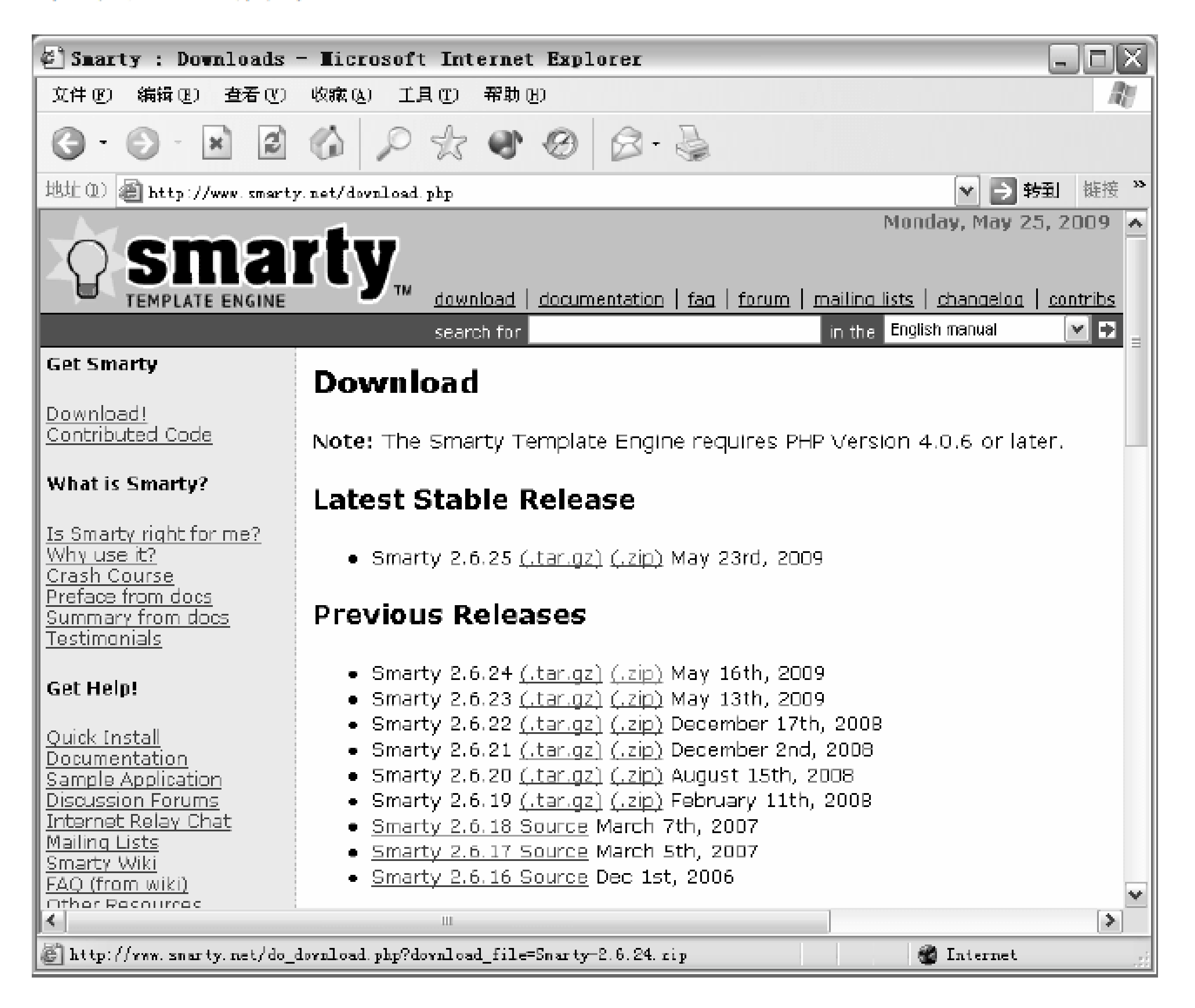


图 17.2　Smarty 模板下载页面

17.1.2　Smarty 模板引擎安装

Smarty 模板引擎的安装步骤如下：

（1）将下载的 Smarty 压缩包解压后，有一个 libs 目录，这里包含了 Smarty 类库的核心文件，包括 smarty.class.php、smarty_Compiler.class.php、config_File.class.php 和 debug.tpl 4 个文件，还有 internals 和 plugins 两个目录，如图 17.3 所示。



图 17.3　Smarty-2.6.23 文件夹

（2）复制 libs 目录到服务器根目录指定的文件夹下。至此 Smarty 安装成功。

17.1.3　Smarty 模板引擎配置

Smarty 模板引擎的配置步骤如下：

（1）确定 Smarty 类库的存储位置。因为 Smarty 类库是通用的，每一个项目都可能会使用到它。所以将 Smarty 存储在根目录下是一个比较理想的选择。这里以将其存储于本堂课根目录 17\sl\17.01\Smarty\文件

夹下为例（为第一个 Smarty 程序配置环境）。

（2）新建 4 个目录 templates、templates_c、configs 和 cache，存储于 17\sl\17.01\Smarty 文件夹下，如图 17.4 所示。



图 17.4　Smarty 目录中的内容

提示：新建的 4 个目录并没有固定的存储位置，只要配置的路径正确，存储在根目录下的任何位置都可以。

（3）创建配置文件。当所有需要的资源都就位之后，下面要做的就是将它们进行整合、调动起来。这就是配置文件的作用，定义服务器、Smarty 的绝对路径，加载 Smarty 类库文件，为创建的 4 个目录设置正确的路径，定义 Smarty 的定界符等操作都可以在配置文件中设置。

通常将配置文件定义到一个单独的文件中，在需要使用时直接通过包含语句调用即可。这里将配置文件存储到 config.php 中，其代码如下：

```
<?php
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);                    //定义服务器的绝对路径
define('SMARTY_PATH','\MR\17\sl\17.01\Smarty\\');                 //定义 Smarty 目录的绝对路径
require BASE_PATH.SMARTY_PATH.'libs\Smarty.class.php';            //加载 Smarty 类库文件
$smarty = new Smarty;                                             //实例化一个 Smarty 对象
$smarty->template_dir = BASE_PATH.SMARTY_PATH.'templates/';       //定义模板文件存储位置
$smarty->compile_dir = BASE_PATH.SMARTY_PATH.'templates_c/';      //定义编译文件存储位置
$smarty->config_dir = BASE_PATH.SMARTY_PATH.'configs/';           //定义配置文件存储位置
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';              //定义缓存文件存储位置
/*   定义定界符   */
$smarty->left_delimiter = '<{';
$smarty->right_delimiter = '}>';
?>
```

上述配置文件的参数说明如下所示。

☑　BASE_PATH：指定服务器的绝对路径。

☑　SMARTY_PATH：指定 Smarty 目录的绝对路径。

☑　require：加载 Smarty 类库文件 Smarty.class.php。

☑　$smarty：实例化 Smarty 对象。

☑　$smarty->template_dir：定义模板目录存储位置。

☑　$smarty->compile_dir：定义编译目录存储位置。

☑　$smarty->config_dir：定义配置文件存储位置。

☑　$smarty->cache_dir：定义模板缓存目录。

☑　$smarty->left_delimiter：定义 Smarty 使用的开始定界符。
☑　$smarty->right_delimiter：定义 Smarty 使用的结束定界符。

提示：有关定界符的使用，开发者可以指定任意的格式，也可以不指定。使用 Smarty 默认的定界符“{”和“}”。

至此，Smarty 的配置讲解完毕。至于将配置文件存储在什么位置，可以根据实际情况而定。

17.1.4　走进 Smarty 模板引擎

了解了 Smarty 模板引擎的下载、安装和配置之后，下面应用 Smarty 模板开发一个实例，通过这个实例体现不出 Smarty 模板开发 Web 程序的优势，但是能让读者了解应用 Smarty 模板开发 Web 程序的流程。

例 17.01　通过 Smarty 模板开发一个实例。（**实例位置：光盘\mr\17\sl\17.01**）

具体步骤如下：

（1）安装 Smarty 模板引擎。将下载的 libs 文件夹存储于服务器根目录下的 17\sl\17.01\Smarty 文件夹中。

（2）在 Smarty 目录下新建 4 个目录，即 templates、templates_c、configs 和 cache，用于存储模板文件、编译文件、配置文件和缓存文件。

（3）配置 Smarty 模板。在根目录 17\sl\17.01 文件夹下创建 config.php 文件，对 Smarty 模板进行配置。配置文件的具体内容请参考 17.1.3 小节。

（4）在 17.01 文件夹下创建 index.php 文件。首先通过 include 语句包含 config.php 文件，对 Smarty 模板引擎进行配置，然后通过 Smarty 中的 assign 方法向模板中传递数据，最后通过 display 方法指定模板页。代码如下：

```
<?php
include("config.php");
/*　使用 Smarty 赋值方法将一对名称/方法发送到模板中　*/
$smarty->assign('title','走进 Smarty 模板引擎');
/*　显示模板　*/
$smarty->display('index.html');
?>
```

（5）在\17.01\Smarty\templates 文件夹下新建一个 index.html 静态页，获取 Smarty 模板变量传递的数据，代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title><{$title}></title>
</head>
<body>
<img src="images/bg.jpg" width="698" height="237" />
</body>
</html>
```

说明：在 index.html 静态页中，大括号“<{}>”为 Smarty 标签的定界符。$title 是从 PHP 动态页中传递的变量。

至此，已经走进 Smarty 模板，本实例的目录结构如图 17.5 所示。

本实例的运行结果如图 17.6 所示。

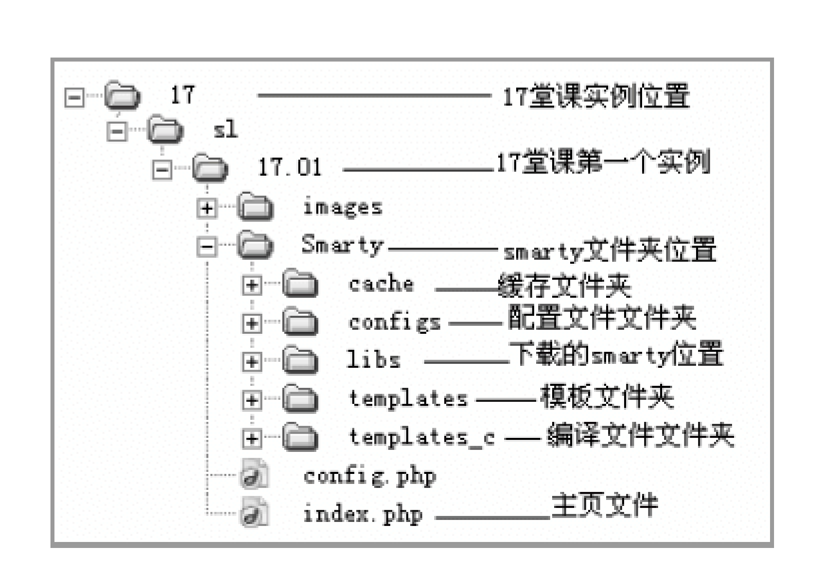


图 17.5　第一个 Smarty 程序的目录结构



图 17.6　第一个 Smarty 程序的运行结果

17.2　Smarty 模板设计——静态页处理

既然 Smarty 可以将用户界面和 PHP 代码实现分离，那么应用 Smarty 模板开发程序也同样包含两部分内容，即 Smarty 模板设计和 Smarty 程序设计。

Smarty 模板设计，其所有操作都是在模板文件中进行的，那么什么是 Smarty 模板文件呢？Smarty 模板文件是由一个页面中所有的静态元素，加上一些定界符“{…}”(Smarty 默认的定界符，读者也可以自行设置定界符）组成的。模板文件统一存储于 templates 目录下（读者可以任意修改这个存储位置，只要路径配置正确即可）。模板文件中不允许出现 PHP 代码段。Smarty 模板中的所有注释、变量、函数等都要包含在定界符内。

17.2.1　基本语法（注释、函数和属性）

在 Smarty 的基本语法中包括 3 项内容，即注释、函数和属性。

1．注释

Smarty 注释包含在两个星号“*”中间，其格式如下：

```
{* 这是注释 *}
```

Smarty 注释不会在模板文件的最后输出中出现，它只是模板内在的注释。

2．函数

每一个 Smarty 标签输出一个变量或者调用某一个函数。在定界符内函数和其属性将被处理和输出。例如：

```
{funcname attr1="val" attr2="val"}.
```

在模板文件中无论是内建函数还是自定义函数都有相同的语法。

内建函数在 Smarty 内部工作，例如，{if}、{section}和{foreach}等，它们只可以使用，不能被修改。

自定义函数通过插件机制起作用，它们是附加函数，读者可以自行编辑、修改和添加，如{assign}和{counter}。



3．属性

大多数函数都有自己的属性以便于明确说明或者修改它们的行为。Smarty 函数的属性类似于 HTML 中的属性。

在定义属性值时，如果是静态数值不需要加引号，如果是字符串建议使用引号，如果属性值是变量，那么也不需要加引号。

17.2.2　Smarty 模板设计变量

Smarty 有几种不同类型的变量。变量的类型取决于它的前缀是什么符号（或者被什么符号包围）。

Smarty 的变量可以直接被输出或者作为函数属性和修饰符（modifiers）的参数，或者用于内部的条件表达式等。

如果要输出一个变量，只需用定界符将它括起来即可。例如：

```
{$title}
{$array[row].id}
<body bgcolor="{#bgcolor#}">
```

1．来自 PHP 页面中的变量

获取 PHP 页面中的变量与在 PHP 中是相同的，也需要使用“$”符号，略有不同的是对数组的读取。在 Smarty 中读取数组有两种方法：一种是通过索引获取，和 PHP 中相似，可以是一维，也可以是多维；另一种是通过键值获取数组元素，这种方法的格式和以前接触过的不太一样，使用符号“.”做连接符。

例如，有一数组$arr=array{'user'=>'明日科技','pass'=>'mrsoft','address'=>'长春市'}，如果要获取 user 的值，表达式的格式应为$arr.user。这个格式同样适用于二维数组。

调用模板内 assign 函数分配的变量也是如此，使用“$”符号加变量名来调用。

2．从配置文件读取变量

Smarty 模板中，在配置文件中也可以定义变量。调用配置文件中变量的格式有以下两种：

☑　使用“#”号，将变量名置于两个“#”号中间，即可像普通变量一样调用配置文件内容。

☑　使用保留变量中的$smarty_config 来调用配置文件。

3．保留变量

在 Smarty 模板中使用保留变量时，无须使用 assign()方法传值，直接调用变量名即可。Smarty 中常用的保留变量如表 17.1 所示。

表 17.1　Smarty 中常用的保留变量

保 留 变 量	说　明
get、post、server、session、cookie、ruquest	等价于 PHP 中的$_GET、$_POST、$_SERVER、$_COOKIE、$_REQUEST
now	当前的时间戳。等价于 PHP 中的 time()
const	用 const 包含修饰的为常量
config	配置文件内容变量

17.2.3 变量调节器

变量调节器作用于变量、自定义函数和字符串。使用“|”符号和调节器名称调用调节器。变量调节器由赋予的参数值决定其行为。参数由“:”符号分开。Smarty 中提供的变量调节器如表 17.2 所示。

表 17.2 Smarty 中提供的变量调节器

名　称	作　用
capitalize	将变量中的所有单词首字母大写
count_characters	计算变量中的字符数。参数值为 TRUE，表示计算空格字符，默认为 FALSE
cat	将 cat 中的值连接到给定的变量后面
count_paragraphs	计算变量中的段落数量
count_sentences	计算变量中句子的数量
count_words	计算变量中的词数
date_format	格式化从函数 strftime()获得的时间和日期。UNIX 或者 MySQL 等的时间戳（parsable by strtotime）都可以传递到 smarty。设计者可以使用 date_format 完全控制日期格式 参数 1 是输出日期的格式，参数 2 是输入为空时默认的时间格式
default	为空变量设置一个默认值。当变量为空或者未分配时，将由给定的默认值替代输出
escape	用于 html 转码、url 转码，在没有转码的变量上转换单引号、十六进制转码、十六进制美化或者 JavaScript 转码。默认是 html 转码 参数 1 指定使用何种编码格式
indent	在每行缩进字符串，默认是 4 个字符。参数 1 作为可选参数，可以指定缩进字符数；参数 2 同样为可选参数，可以指定缩进用什么字符代替。需要注意的是，如果在 HTML 中使用缩进，那么需要使用 (空格)来代替缩进，否则没有效果
lower	将变量字符串转换为小写
nl2br	所有的换行符将被替换成 ，功能同 PHP 中的 nl2br()函数相同
regex_replace	寻找和替换正则表达式。参数 1 指定用来替换的文本字串
replace	搜索和替换字符串。参数 1 是将被替换的文本字串，参数 2 是用来替换的文本字串
spacify	插空，在字符串的每个字符之间插入空格或者其他的字符（串）。参数 1 指定将在两个字符之间插入的字符（串）
string_format	字符串的格式化。采用 sprintf()函数的语法。参数 1 指定使用的格式化方式
strip	用一个空格或一个给定字符替换所有重复空格、换行和制表符。需要注意的是，如果要去除模板文本中的区块，请使用 strip 函数
strip_tags	去除“<”和“>”标签，包括“<”和“>”之间的任何内容
truncate	从字符串开始处截取指定长度的字符，默认是 80 个字节 参数 1 设置截取字符的数量，参数 2 设置截取后追加在截取词后面的字符串，参数 3 设置是截取到词的边界（FALSE）还是精确到字符（TRUE）
upper	将变量转换为大写
wordwrap	控制段落的宽度（也就是多少个字符一行，超过这个字符数换行），默认是 80 个字节 参数 1 设置段落（句子）的宽度，参数 2 设置使用什么字符进行约束（默认是换行符\n），参数 3 设置是约束到词的边界（FALSE）还是精确到字符（TRUE）

技巧：如果给数组变量应用单值变量的调节，结果是数组的每个值都被调节。如果只想调节器用一个值调节整个数组，那么必须在调节器名字前加上@符号，如{$articleTitle|@count}（这将会在$articleTitle 数组中输出元素的数目）。

17.2.4　内建函数（动态文件、模板文件的包含和流程控制语句）

Smarty 自身定义了一些内建函数，存储于 Smarty 模板中。它是模板语言的一部分，用户不能创建名称和内建函数相同的自定义函数，也不能修改内建函数。内建函数包括 foreach、if 和 section 此类的流程控制语句，还包括像 include 和 include_php 这样的函数。本小节将讲解一些常用内建函数的使用方法，如果读者要了解全部内建函数的知识，可以参考 Smarty 手册。

1．foreach 循环控制

Smarty 模板中的 foreach 语句可以循环输出数组。与另一个循环控制语句 section 相比，在使用格式上要简单得多，一般用于简单数组的处理。foreach 语法如下：

```
{foreach name=foreach_name key=key item=item from=arr_name}
…
{/foreach}
```

参数 name 为该循环的名称，参数 key 为当前元素的键值，参数 item 是当前元素的变量名，参数 from 是该循环的数组。其中，item 和 from 是必要参数，不可省略。

2．include 函数——在模板中包含子模板

include 函数用于在当前模板中包含其他模板，当前模板中的变量在被包含的模板中可用。该函数的语法如下：

```
{include file="file_name " assign=" " var="   "}
```

参数 file 指定包含模板文件的名称，为必选参数；参数 assign 指定一个变量保存包含模板的输出；参数 var 传递给待包含模板的本地参数，只在待包含模板中有效。

3．If…elseif…else 条件语句

if 条件语句的使用和 PHP 中的 if 大同小异。需要注意的一点是 if 必须以/if 为结束标记。其语法格式如下：

```
{if 条件语句 1}
    语句 1
{elseif 条件语句 2}
    语句 2
{else}
    语句 3
{/if}
```

在上述的条件语句中，除了可以使用 PHP 中的<、>、=、!=等常见运算符外，还可以使用 eq、ne、neq、gt、lt、lte、le、gte、ge、is even、is odd、is not even、is not odd、not、mod、div by、even by、odd by 等修饰词修饰。

4．ldelim 和 rdelim——输出大括号“{”和“}”

ldelim 和 rdelim 用于输出定界符，也就是大括号“{”和“}”。因为模板引擎总是尝试解释大括号内的

内容，因此如果需要输出大括号，则可以使用这两个函数。

例如，在模板页面中输出一个 JavaScript 脚本，因为 JavaScript 脚本中会涉及大括号的使用，所以应用 ldelim 和 rdelim 输出 JavaScript 脚本中的大括号。代码如下：

```
<script language=javascript>
function check_form() {ldelim}
	if (user.value == ''){ldelim}
		alert('请输入用户名');
		return false;
	{rdelim}
{rdelim}
</script>
```

你问我答： 通过 literal 标签输出大括号。

通过 ldelim 和 rdelim 标签可以输出 JavaScript 脚本中的大括号，这个方法需要对每个大括号都进行操作。如果使用 literal 标签就没有那么麻烦了，它可以将整个标签区域内的数据当作文本处理。同样是在模板文件中输出 JavaScript 脚本，应用 literal 标签就简单多了，代码如下：

```
{literal}
<script language=javascript>
function check_form() {
	if (user.value == ''){
		alert('请输入用户名');
		return false;
	}
}
</script>
{/literal}
```

技巧： 如果要在 Smarty 模板文件中直接输出 JavaScript 脚本或者定义 CSS 样式，并且 Smarty 使用默认的定界符“{}”，那么就会应用到上述两个函数中的一个，对 JavaScript 脚本或者 CSS 样式中的大括号进行输出。

5．section 循环控制

section 是 Smarty 模板中的另一个循环语句，该语句可用于比较复杂的数组。section 的语法结构如下：

```
{section name="sec_name"loop=$arr_name start=num step=num max= show=}
```

section 语句的参数说明如表 17.3 所示。

表 17.3　section 语句的参数说明

参　数	说　明
name	循环的名称
loop	循环的数组
start	表示循环的初始位置，如 start=2 说明循环是从 loop 数组的第二个元素开始
step	表示步长，如 step=2，那么循环一次后数组的指针将向下移动两位，依此类推
max	设定循环最大执行次数
show	决定是否显示该循环

section 循环语句最擅长的是操作 ADODB 从数据库中读取到的数据，因为 ADODB 返回的数据就是一个二维数组。

17.2.5　自定义函数

Smarty 中包含很多自定义函数，通过这些自定义函数可以实现很多的功能。Smarty 中的自定义函数如表 17.4 所示。

表 17.4　Smarty 中的自定义函数

名　　称	作　　用
assign	用于在模板被执行时为模板变量赋值。参数 var 是被赋值的变量名，参数 value 是赋给变量的值
counter	用于输出一个计数过程。counter 保存了每次计数时的当前计数值
cycle	用于轮转使用一组值。该特性使得在表格中交替输出颜色或轮转使用数组中的值变得很容易参数 name 指定轮转的名称；参数 values 指定待轮转的值，可以是用逗号分隔的列表（请查看 delimiter 属性）或一个包含多值的数组；参数 print 设置是否输出值；参数 advance 设置是否使用下一个值（为 FALSE 时使用当前值）；参数 delimiter 设置 values 属性中使用的分隔符，默认是逗号；参数 assign 指定输出值将被赋给模板变量的名称
debug	将调试信息输出到页面上。该函数是否可用取决于 Smarty 的 debug 设置
eval	按处理模板的方式获取变量的值。该特性可用于在配置文件的标签/变量中嵌入其他模板标签/变量
fetch	用于从本地文件系统、HTTP 或 FTP 上取得文件并显示文件的内容。如果文件名称以"http://"开头，将取得该网站页面并显示；如果文件名称以"ftp://"开头，将从 ftp 服务器取得该文件并显示
html_checkboxes	根据给定的数据创建复选框组
html_image	创建一个图像的 HTML 标签。如果没有提供高度和宽度值，将根据图像的实际大小自动取得
html_options	根据给定的数据创建选项组
html_radios	根据给定的数据创建单选按钮组
html_select_date	创建日期下拉菜单。它可以显示任意年月日
html_select_time	创建时间下拉菜单。它可以显示任意时分秒
html_table	将数组中的数据填充到 HTML 表格中
math	允许模板设计者在模板中进行数学表达式运算
mailto	mailto 自动生成电子邮件链接，并根据选项决定是否对地址信息编码
popup	用于创建 JavaScript 弹出窗口
textformat	用于格式化文本。该函数主要清理空格和特殊字符

17.2.6　配置文件

配置文件的应用，有利于设计者管理文件中的模板全局变量。例如，定义一个模板色彩变量。一般情况下如果想改变一个程序的外观色彩，必须更改每一个文件的颜色变量。如果有配置文件，色彩变量就可以保存在一个单独的文件中，只要改变配置文件就可以实现色彩的更新。

1．创建配置文件

对于配置文件可以任意命名，其存储位置由 Smarty 对象的$config_dir 属性指定。如果存在不只在一个

区域内使用的变量值，可以使用三引号（"""）将它完整地封装起来。在创建配置文件时，建议在程序运行前使用“#”加一些注释信息，这样有助于程序的阅读、更新。

在配置文件中既可以声明全局变量，也可以声明局部变量。如果声明局部变量，可以使用中括号“[]”括起来，在中括号之内声明的变量属于局部变量，而中括号之外声明的变量都是全局变量。中括号的使用不仅使配置文件中声明变量的模块变得清晰，而且可以在模板中选择加载中括号内的变量。

例如，创建一个配置文件，分别声明全局变量和局部变量。代码如下：

```
# global variables                     #在每行之前使用#，表示注释
title = "引用配置文件"                 #声明全局变量
[table]                                #声明局部变量
border = "1"
cellpadding="1"
cellspacing="1"
bordercolor="#FFFFFF"
table_bgcolor="#333333"
[td]                                   #声明局部变量
bgcolor="#FFFFFF"
```

如果某个特定的局部变量已经载入，则全局变量和局部变量都可以载入。如果当某个变量名既是全局变量又是局部变量时，局部变量将被优先赋予值来使用。如果在一个局部中两个变量名相同，最后一个将被赋值使用。

2．加载配置文件

加载配置文件应用 Smarty 的内建函数 config_load，其语法如下：

```
{config_load file="file_name " section="add_attribute" scope="" global=""}
```

config_load()函数的参数说明如表 17.5 所示。

表 17.5　config_load()函数的参数说明

参　数	说　明
file	指定包含的配置文件的名称
section	附加属性，当配置文件中包含多个部分时应用，指定具体从哪一部分中取得变量
scope	加载数据的作用域，取值必须为 local、parent 或 global。local 说明该变量的作用域为当前模板，parent 说明该变量的作用域为当前模板和当前模板的父模板（调用当前模板的模板），global 说明该变量的作用域为所有模板。当指定 scope 属性时，可以设置 global 属性，但模板忽略该属性值，而以 scope 属性为准
global	说明加载的变量是否全局可见，等同于 scope=parent

3．引用配置文件中的变量

配置文件加载成功后，就可以在模板中引用配置文件中声明的变量了。引用配置文件应用的是“#”或者 Smarty 的保留变量$smarty.config。其应用示例如下：

```
{ config_load file="file_con.conf"}                    {* 加载配置文件 *}
{#title#}
    <td height="228" colspan="2" align="left" valign="top" class="{$smarty.config.styles}">
```

17.3　Smarty 程序设计——动态文件操作

Smarty 程序设计在动态 PHP 文件中进行操作，其功能可以分为两种：一种功能是配置 Smarty，如变量 template_dir、$config_dir 等；另一种功能是和 Smarty 模板之间的交互，如方法 assign、display。

17.3.1　SMARTY_PATH 常量

SMARTY_PATH 常量定义 Smarty 类文件的完整系统路径，如果没有定义 Smarty 目录，Smarty 将会试着自动创建合适的值。如果定义了路径必须要以斜线结束。该常量的应用是在 Smarty 的配置文件中，通过它获取 Smarty 类的绝对路径。

例如，在 17.1.3 小节创建的配置 config.php 中，就应用到这个常量。关键代码如下：

```
define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);                    //定义服务器的绝对路径
define('SMARTY_PATH','\MR\17\sl\17.01\Smarty\\');                 //定义 Smarty 目录的绝对路径
require BASE_PATH.SMARTY_PATH.'libs\Smarty.class.php';             //加载 Smarty 类库文件
```

17.3.2　Smarty 程序设计变量

在 Smarty 中提供了很多的变量，这里只讲解比较常用的几个，如果读者想详细地了解 Smarty 变量，请参考 Smarty 的手册。

- ☑ $template_dir：模板目录。该目录用来存放 Smarty 模板，在前面的实例中，所有的.html 文件都是 Smarty 模板。模板的后缀没有要求，一般为.html、.tpl 等。
- ☑ $compile_dir：编译目录。顾名思义，就是编译后的模板和 PHP 程序所生成的文件默认路径为当前执行文件所在的目录下的 templates_c 目录。进入到编译目录，可以发现许多“%%…%%index.html.php”格式的文件。随便打开一个这样的文件可以发现，实际上 Smarty 将模板和 PHP 程序又重新组合成一个混编页面。
- ☑ $cache_dir：缓存目录。该目录用来存放缓存文件。同样，在 cache 目录下可以看到生成的.html 文件。如果 caching 变量开启，那么 Smarty 将直接从这里读取文件。
- ☑ $config_dir：配置目录。该目录用来存放配置文件。
- ☑ $debugging：调试变量。该变量可以打开调试控制台。只要在配置文件（config.php）中将 $smarty->debugging 设为 TRUE 即可使用。
- ☑ $caching：缓存变量。该变量可以开启缓存。只要当前模板文件和配置文件未被改动，Smarty 就直接从缓存目录中读取缓存文件而不重新编译模板。

17.3.3　Smarty 方法

在 Smarty 提供的方法中，最常用的是 assign 方法和 display 方法。

1．assign 方法

assign 方法用于在模板被执行时为模板变量赋值。语法如下：

```
{assign var=" " value=" "}
```

参数 var 是被赋值的变量名，参数 value 是赋给变量的值。

2．display 方法

display 方法用于显示模板，需要指定一个合法的模板资源的类型和路径。还可以通过第二个可选参数指定一个缓存号，相关的信息可以查看缓存。

```
void display (string template [, string cache_id [, string compile_id]])
```

参数 template 指定一个合法的模板资源的类型和路径；参数 cache_id 为可选参数，指定一个缓存号；参数 compile_id 为可选参数，指定编译号。编译号可以将一个模板编译成不同版本使用，如针对不同的语言编译模板。编译号的另外一个作用是，如果存在多个$template_dir 模板目录，但只有一个$compile_dir 编译后存档目录，这时可以为每一个$template_dir 模板目录指定一个编译号，以避免相同的模板文件在编译后会互相覆盖。相对于在每一次调用 display()时都指定编译号，也可以通过设置$compile_id 编译号属性来一次性设定。

17.3.4 Smarty 缓存

在讲解 Smarty 的缓存之前，先将它和 Smarty 的编译过程做一个对比，让读者明白缓存到底意味着什么。

Smarty 的编译功能默认是开启的，而 Smarty 缓存则必须由开发人员来开启。

编译的过程是将模板转换为 PHP 脚本，虽然在模板没有被修改的情况下，不会重新执行转换过程，但这个编译过的模板其实就是一个 PHP 脚本，只是减少了模板转换的压力，仍需要在逻辑层执行获取数据的操作，而这个获取数据的操作是耗费内存最大的。

缓存则不仅将模板转换为 PHP 脚本，而且将模板内容转换为静态页面，不仅减少了模板转换的压力，也不再需要在逻辑层执行获取数据的操作。

这就是 Smarty 的缓存机制，它是一种更加理想的开发 Web 程序的方法。下面就来学习这种技术。

1．创建缓存

开启缓存的方法非常简单，只要将 Smarty 对象中$config 的值设置为 TRUE 即可，同时还要通过 Smarty 对象中的$cache_dir 属性指定缓存文件的存储位置。操作代码如下：

```
$smarty->caching=true;                                      //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';        //定义缓存文件存储位置
```

2．缓存的生命周期

缓存创建成功后，必须为它设置一个生命周期，如果它一直不更新，那么就没有任何意义。设置缓存生命周期应用的是 Smarty 对象中的$cache_lifetime 属性，缓存时间以秒为单位，默认值是 3600 秒。操作代码如下：

```
$smarty->caching=true;                                      //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';        //定义缓存文件存储位置
$smarty->cache_lifetime=3600                                //设置缓存时间为 1 小时
```

如果将$caching 的值设置为 2，那么就可以控制单个缓存文件各自的过期时间。

3．同一模板生成多个缓存

在实际的程序开发中，经常会遇到这样的情况，同一个模板文件生成多个页面。而此时要对多个页面进行缓存，应用的是 Smarty 中的 display()方法，通过该方法的第二个参数设置缓存号，有几个不同的缓存



号就有几个缓存页面。操作代码如下：

```
$smarty->caching=true;                                    //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';       //定义缓存文件存储位置
$smarty->cache_lifetime=3600;                             //设置缓存时间为 1 小时
$smarty->display('index.html',$_GET['id']);               //将 id 作为第二个参数传递
```

4．判断模板文件是否已被缓存

如果页面已经被缓存，那么就可以直接调用缓存文件，而不再执行动态获取数据和输出的操作。为了避免在开启缓存后，再次执行动态获取数据和输出操作给服务器带来的压力，最佳的方法就是应用 Smarty 对象中的 is_cached()方法，判断指定的模板是否存在缓存，如果存在则直接执行缓存中的文件，否则执行动态获取数据和输出的操作。操作代码如下：

```
$smarty->caching=true;                                    //开启缓存
if(!$smarty->is_cached('index.html')){
    //执行动态获取数据和输出的操作
}
$smarty->display('index.html');
```

你问我答：如何判断同一模板中的多个缓存文件？

判断同一模板中的多个缓存是否存在与同一模板生成多个缓存类似，都是以缓存号为依据。判断同一模板的多个缓存是否存在应用 is_cached()方法，通过该方法的第二个参数设置缓存号，判断对应的缓存是否存在。其方法如下：

```
$smarty->caching=true;                                    //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';       //定义缓存文件存储位置
$smarty->cache_lifetime=3600;                             //设置缓存时间为 1 小时
if(!$smarty->is_cached('index.html',$_GET['id'])){
    //执行动态获取数据和输出的操作
}
$smarty->display('index.html',$_GET['id']);               //将 id 作为第二个参数传递
```

5．清除模板中缓存

缓存的清除有两种方法，分别介绍如下。

（1）clear_all_cache()方法，清除所有模板缓存。其语法如下：

```
void clear_all_cache (int expire time)
```

可选参数 expire time，可以指定一个以秒为单位的最小时间，超过这个时间的缓存都将被清除。

（2）clear_cache()方法，清除指定模板的缓存。其语法如下：

```
void clear_cache (string template [, string cache id [, string compile id [, int expire time]]])
```

如果这个模板有多个缓存，可以用第二个参数指定要清除缓存的缓存号，还可以通过第 3 个参数指定编译号。可以把模板分组，以便可以方便地清除一组缓存。第 4 个参数是可选的，用来指定超过某一时间（以秒为单位）的缓存才会被清除。

例如，分别应用这两种方法清除缓存。代码如下：

```
$smarty->caching=true;                               //开启缓存
$smarty->clear_all_cache();                          //清除所有缓存
$smarty->clear_cache('index.html');                  //清除 index.html 模板的缓存
$smarty->clear_cache('index.html','$_GET['id']');    //清除 index.html 模板中一个指定缓存号的缓存
$smarty->display('index.html');                      //指定模板页
```

17.4 照猫画虎——基本功训练

17.4.1 基本功训练 1——封装 Smarty 模板的配置方法

视频讲解：光盘\mr\17\lx\封装 Smarty 模板的配置方法.exe

实例位置：光盘\mr\17\zmhh\01

Smarty 模板的配置方法是应用 Smarty 模板的关键，但是用户不能在每次使用 Smarty 时都写一遍配置文件。这样不仅使服务器增加了负担，更主要的是使程序的可读性变差，为了解决这个问题笔者决定将 Smarty 的配置方法封装到一个类中，这样可以在需要应用 Smarty 模板的地方直接调用类文件即可。在本实例中，通过封装的类文件完成 Smarty 模板的配置，开发一个简单的实例，运行结果如图 17.7 所示。



图 17.7　第一个 Smarty 程序的运行结果

实现步骤如下：

（1）下载和安装 Smarty 类库，本项目中将使用的 Smarty 类库文件夹 libs 存储于本堂课的 17\zmhh\下。

（2）在本堂课 17\zmhh\文件夹下，创建 01\system\Smarty 文件夹，新建模板文件夹（templates）、编译文件夹（templates_c）、配置文件夹（configs）和缓存文件夹（cache）。

（3）将 Smarty 配置方法封装到类中，并存储到 17\zmhh\01\system\system.smarty.inc.php 文件中。首先，包含 Smarty 类文件 Smarty.class.php。然后，定义 SmartyProject 类，继承 Smarty 父类。最后，定义 SmartyProject() 方法，设置 Smarty 中模板文件（templates）、编译文件（templates_c）、配置文件（configs）和缓存文件（cache）的存储位置。配置文件 system.smarty.inc.php 的代码如下：

```
<?php
require("../libs/Smarty.class.php");              //调用 Smarty 文件
class SmartyProject extends   Smarty{             //定义类，继承 Smarty 父类
      function SmartyProject(){                   //定义方法，配置 Smarty 模板
            $this->template_dir = "./";           //指定模板文件存储在根目录下
            $this->compile_dir = "./system/Smarty/templates_c/";      //指定编译文件存储位置
            $this->config_dir = "./system/Smarty/configs/";
            $this->cache_dir = "./system/Smarty/cache/";
      }
}
?>
```

（4）既然已经将 Smarty 的配置方法存储到一个类中，那么就需要对类进行实例化，根据返回的对象名称调用 Smarty 中的方法，类的实例化操作在 17\zmhh\01\system\system.inc.php 文件中完成，其返回对象名为$smarty。代码如下：

```
<?php
      require("system.smarty.inc.php");           //调用类文件
      $smarty=new SmartyProject();                //执行类的实例化操作
?>
```

（5）在 17\zmhh\01\文件夹下创建 index.php 文件。首先通过 include_once 语句包含 system\system.inc.php 配置文件，对 Smarty 模板引擎进行配置，然后通过 Smarty 中的 assign 方法向模板中传递数据，最后通过 display 方法指定模板页。代码如下：



```
<?php
include_once("system/system.inc.php");
/*  使用 Smarty 赋值方法将一对名称/方法发送到模板中   */
$smarty->assign('title','走进 Smarty 模板引擎');
/*  显示模板  */
$smarty->display('index.html');
?>
```

（6）在 17\zmhh\01\文件夹下新建一个 index.html 静态页，获取 Smarty 模板变量传递的数据。代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/ xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>{$title}</title>
</head>
<body>
<img src="images/bg.jpg" width="698" height="237" />
</body>
</html>
```

照猫画虎：在封装的 Smarty 配置类中定义 smarty 的定界符为“<{”和“}>”。提示：应用 left_delimiter 和 right_delimiter。（20 分）（实例位置：光盘\mr\17\zmhh\01_zmhh）

17.4.2　基本功训练 2——Smarty 模板中的页面设计

视频讲解：光盘\mr\17\lx\Smarty 模板中的页面设计.exe

实例位置：光盘\mr\17\zmhh\02

在 Smarty 模板页中同样可以将网页的头文件、尾文件和主文件进行分别存储，从而达到页面重用的效果，这就是本实例要讲解的内容。在 Smarty 模板页中设计网页页面，运行结果如图 17.8 所示。



图 17.8　Smarty 模板中的页面设计

操作步骤如下：

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，这里不再赘述。

（2）创建 index.php 文件，应用 switch 语句根据超链接传递的变量值，包含不同的 PHP 脚本文件，同时将对应的模板页名称赋给指定的模板变量。代码如下：

```
<?php
require("system/system.inc.php");                    //包含 Smarty 模板的配置类
if(isset($_GET['caption'])){                         //判断超链接的变量值
    $caption=$_GET['caption'];
}else{
    $caption="";
}
switch ($caption){                                   //根据 switch 语句实现不同页面之间的调整
    case "reg";
        include "reg.php";                           //包含 PHP 文件
        $smarty->assign('admin_phtml','reg.html');   //将对应的静态页文件赋给指定的模板变量
    break;
    case "log";
        include "log.php";
        $smarty->assign('admin_phtml','log.html');
    break;
    case "from";
        include "for.php";
        $smarty->assign('admin_phtml','for.html');
    break;
    case "mes";
        include "mes.php";
        $smarty->assign('admin_phtml','mes.html');
    break;
    default:
        include "for.php";
        $smarty->assign('admin_phtml','for.html');
    break;
}
$smarty->assign("title","Smarty 模板中的页面设计--".$caption);        //创建模板变量
$smarty->display("index.html");                                     //指定模板页
?>
```

（3）创建 index.html 模板页，应用 include 函数载入页面的头文件（top.html）、尾文件（bottom.html）和主文件。关键代码如下：

```
{$title}
{include file=top.html}
{include file=$admin_phtml}
{include file=right.html}
{include file=bottom.html}
```

（4）创建 include 函数加载的模板文件和动态 PHP 文件。

照猫画虎：在 Smarty 模板中直接定义 CSS 样式。提示：应用{literal}和{/literal}。（20 分）（实例位置：光盘\mr\17\zmhh\02_zmhh）

17.4.3　基本功训练 3——if 语句判断当前用户权限

视频讲解：光盘\mr\17\lx\if 语句判断当前用户权限.exe

实例位置：光盘\mr\17\zmhh\03

在动态 PHP 文件中，可以通过 SESSION 变量的值判断当前用户是否具有访问权限，而在 Smarty 模板



页中不可以使用 SESSION 变量，那么该如何判断用户是否具有访问权限呢？这就是在本实例中将要讲解的内容。应用 if 语句在 Smarty 模板页中判断用户是否具有访问权限，运行结果如图 17.9 所示。



图 17.9　if 语句判断用户是否具备访问权限

具体步骤如下：

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，创建存储编译文件、缓存文件和配置文件的文件夹。其具体内容可以参考 17.4.1 小节，这里不再赘述。

（2）创建 index.php 文件，包含 Smarty 配置文件，指定 Smarty 的模板页 index.html。

（3）创建 Smarty 模板页 index.html，设计用户登录页面，添加用户登录的表单元素，并且将用户的登录信息提交到 index_ok.php 文件。

（4）创建 index_ok.php 文件，获取表单中提交的数据，判断提交的用户名和密码是否正确，如果正确则通过 display()方法指定模板页 main.html，并且为模板变量 competence 赋值为 T；否则指定到模板页 main.html，为模板变量 competence 赋值为 F。关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include("system/system.inc.php");                                       //包含配置文件
if($_POST['Submit']!="" && $_POST['user']!="" && $_POST['pass']!=""){
        if($_POST['user']=="mr" && $_POST['pass']=="mrsoft"){
            $smarty->assign("competence","T");                          //为模板变量赋值
            $smarty->display('main.html');                              //指定模板页
        }else{
            $smarty->assign("competence","F");
            $smarty->display('main.html');
        }
}else{
        echo "<script>alert('用户名和密码不能为空！'); window.location.href='index.php';</script>";
}
?>
```

（5）创建 main.html 模板页。首先应用 if 语句对模板变量的值进行判断，如果值为 T 则输出本页内容，如果值为 F 则给出提示信息。关键代码如下：

```
{if $competence=="T"}
//省略了部分内容
{/if}
{if $competence=="F"}
<script>alert('您没有权限访问，请重新登录');window.location.href="index.php";</script>
{/if}
```

照猫画虎：在 Smarty 模板中生成数字验证码。（20 分）（实例位置：光盘\mr\17\zmhh\03_zmhh）

17.4.4　基本功训练 4——Smarty 模板中日期、时间的格式化输出

视频讲解：光盘\mr\17\lx\Smarty 模板中日期、时间的格式化输出.exe
实例位置：光盘\mr\17\zmhh\04

在 PHP 脚本中日期、时间的格式化输出最常用的就是 date()函数，那么在 Smarty 模板中该如何完成日期、时间的输出呢？这就是本实例中要讲解的内容。通过 Smarty 模板中的 date_format 函数完成日期、时间的格式化输出，其运行结果如图 17.10 所示。



图 17.10　Smarty 模板中日期的格式化输出

具体步骤如下：

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，其具体内容可以参考 17.4.1 小节，这里不再赘述。

（2）创建 index.php 文件，包含 Smarty 配置文件，指定 Smarty 的模板页 index.html。

（3）创建 Smarty 模板页 index.html，在 index.html 模板文件中插入 date_format 函数，输出系统的当前时间。关键代码如下：

```
当前时间：  {$smarty.now|date_format:" %A   %B - %e - %Y"}
<!--{$smarty.now|date_format}
{$smarty.now|date_format:"%A, %B %e, %Y":$times}
{$smarty.now|date_format:"%H:%M:%S"}
{$yesterday|date_format}
{$yesterday|date_format:"%A, %B %e, %Y"}
-->
```

照猫画虎：制作日期时间选择器。提示：应用{html_select_date}和{html_select_time use_24_hours=true}。（20 分）（实例位置：光盘\mr\17\zmhh\04_zmhh）

17.4.5　基本功训练 5——register_object 方法注册模板对象

视频讲解：光盘\mr\17\lx\register_object 方法注册模板对象.exe
实例位置：光盘\mr\17\zmhh\05

在 PHP 中，可以通过 function 创建自定义函数，完成一些特殊的操作，同样在 Smarty 中可以通过 register_object 方法注册对象，在模板页中完成一些特殊的操作。在本训练中通过 register_object 方法注册对象，定义两个方法 moneyFormat()和 unHtml()，分别对留言板中的评分进行转义，保留两位小数；对留言内容中的 html 标记进行转义，其运行结果如图 17.11 所示。

具体步骤如下：

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录；创建类文件 system.class.inc.php，定义数据库连接、操作类。创建 system.smarty.inc.php 文件，封装 Smarty 的配置类，定义类的实例化文件 system.inc.php，完成各个类的实例化操作，并返回操作对象。

（2）创建 index.php 文件，包含类的实例化操作文件；创建类，定义评分转义和 html 标记转义的方法，通过 register_object 注册模板对象；完成数据库中数据的读取，最终指定模板页。代码如下：

```
<?php
require_once("system/system.inc.php");              //调用指定的文件
```



```
class Util {
	function moneyFormat($integral) {			//将数字转换成有两位小数的形式
		return str_replace ( ',', '', number_format ( $integral, 2 ) );
	}
	function unHtml($text) {				//转换字符串中的 html 标签
		$str = htmlspecialchars ( $text );
		$str = ereg_replace ( '<br>', '\n', $str );
		$str = nl2br ( $str );
		$str = ereg_replace ( ' ', ' ', $str );
		return $str;
	}
}
$ntil=new Util();					//实例化对象
$smarty->register_object("Util", $ntil,null,false);	//注册模板对象
$array=$admindb->ExecSQL("select * from tb_guestbook order by id desc limit 2",$conn);	//查询数据库中数据
if(!$array){
	$smarty->assign("iscommo","F");			//判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
	$smarty->assign("iscommo","T");			//判断如果执行成功，则输出模板变量 iscommo 的值为 T
	$smarty->assign("array",$array);
}
	$smarty->assign('title','register_object()方法注册模板对象');
	$smarty->display('index.html');

?>
```



图 17.11　通过 register_object 方法注册对象

（3）创建 index.html 模板页，输出模板变量中传递的数据，并且调用注册模板对象中的方法，对评分和 html 标记进行转换操作。同时创建一个 form 表单，用于提交留言信息，将数据提交到 index_ok.php 文件中。

（4）创建 index_ok.php 文件，将表单中提交的留言信息存储到指定的数据表中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );	//设置文件编码格式
require_once("system/system.inc.php");		//调用指定的文件
```

```
$title=$_POST['title'];                              //获取标题
$content=$_POST['content'];                          //获取内容
$integral=$_POST['integral'];
$createtime=date("Y-m-j H:i:s");                     //定义时间
$arr=$admindb->ExecSQL("insert into tb_guestbook(title,content,createtime,integral) values ('$title','$content',
'$createtime','$integral')",$conn);                  //执行添加语句
if($arr){
	echo "<script>alert('留言发表成功!');window.location.href='index.php';</script>";
}else{
	echo "<script>alert('留言发表失败!');history.back();</script>";
}
?>
```

照猫画虎：通过 register_function 方法注册模板函数，完成留言信息的输出。（20 分）（实例位置：光盘\mr\17\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

17.5 情景应用——拓展与实践

17.5.1 情景应用 1——Smarty 模板中的 truncate 方法截取字符串

视频讲解：光盘\mr\17\lx\Smarty 模板中的 truncate 方法截取字符串.exe

实例位置：光盘\mr\17\qjyy\01

在 Smarty 中，提供 truncate 方法对字符串进行截取，该方法会截取到一个词的末尾。在本实例中，使用 truncate 方法对论坛中标题和内容进行截取，并且应用省略号替换截取的内容，其运行结果如图 17.12 所示。



图 17.12 truncate 方法截取字符串

具体步骤如下：

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录；创建类文件 system.class.inc.php，定义数据库连接、操作类。创建 system.smarty.inc.php 文件，封装 Smarty 的配置类，定义类的实例化文件 system.inc.php，完成各个类的实例化操作，并返回操作对象。

（2）创建 index.php 文件，载入类的实例化文件；将从数据库中读取的数据存储到模板变量中，最终指定模板页。代码如下：

```
<?php
require_once("system/system.inc.php");           //调用指定的文件
$array=$admindb->ExecSQL("select * from tb_guestbook order by id desc limit 8",$conn);
if(!$array){
      $smarty->assign("iscommo","F");             //判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
      $smarty->assign("iscommo","T");             //判断如果执行成功，则输出模板变量 iscommo 的值为 T
     $smarty->assign("array",$array);
}
$smarty->assign('title','truncate 方法截取字符串');
$smarty->display('index.html');

?>
```

（3）创建 index.html 模板页，获取模板变量传递的数据，通过 section 循环输出数据，并且通过 truncate 方法对输出的标题和内容进行截取操作，标题截取 30 个字节，内容截取 60 个字节，应用省略号补齐截取部分。关键代码如下：

```
{section name=id loop=$array}
<tr>
<td>
{$array[id].id}{$array[id].title|truncate:30:"...":false};
积分：{$array[id].integral}
时间：{$array[id].createtime|truncate:12:""}<br>
            {$array[id].content|truncate:60:"...":false}
</td>
</tr>
{/section}
```

DIY：关键字描红。提示：应用 Smarty 模板中的 replace 方法。（20 分）（实例位置：光盘\mr\17\qjyy\01_diy）

17.5.2　情景应用 2——Smarty 模板制作后台管理系统主页

视频讲解：光盘\mr\17\lx\Smarty 模板制作后台管理系统主页.exe

实例位置：光盘\mr\17\qjyy\02

在本实例中开发一个后台管理系统，包括管理员的登录、退出，后台管理系统中各个模块的功能展示，当然这些模块都只是简单的架构，没有实现具体的功能。因为开发本实例的主要目的是让读者了解如何通过 Smarty 模板构建后台管理系统的主页，而非某些具体功能模块的开发。本实例首先展示后台管理的登录模块，如图 17.13 所示；登录成功后将进入到后台管理系统的主页，如图 17.14 所示。



图 17.13　后台登录



图 17.14　后台管理主页

具体步骤如下：

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录；创建类文件 system.class.inc.php，定义数据库连接、操作类。创建 system.smarty.inc.php 文件，封装 Smarty 的配置类，定义类的实例化文件 system.inc.php，完成各个类的实例化操作，并返回操作对象。

（2）创建 index.html 文件，设计管理员登录页面；编写 login_ok.php 文件，完成管理员的登录操作。

（3）创建 main.php 文件。首先，通过 header()函数设置页面的编码格式，并初始化 SESSION 变量。然后，根据 SESSION 变量判断当前用户是否具有访问权限。接着，如果具有访问权限，则应用 swicth 语句，根据超链接传递的参数值，完成在不同页面之间的跳转操作，即通过 include 语句包含不同的动态 PHP 文件，同时将动态 PHP 文件对应的模板文件名称赋给模板变量。最后，指定模板页 main.html。代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );            //设置文件编码格式
session_start();                                                //初始化 SESSION 变量
require("system/system.inc.php");
if(isset($_GET['caption'])){
    $caption=$_GET['caption'];
}else{
    $caption="";
}
if($_SESSION['user']!="" and $_SESSION['pass']!=""){            //判断用户是否具有访问权限
switch ($caption){                                              //完成在不同模块之间的跳转操作
    case "商品添加";
        include "sho_insert.php";                               //包含 PHP 脚本文件
        $smarty->assign('admin_phtml','sho_insert.html');
                                        //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
//省略了部分代码
    default:
        include "sho_update.php";
        $smarty->assign('admin_phtml','sho_update.html');
    break;
}
$smarty->assign("title","后台管理系统--".$caption);
$smarty->assign("caption",$caption);
$smarty->assign("type",$_GET['type']);
$smarty->assign("dates",date("Y 年 m 月 d 日"));
$smarty->assign("user",$_SESSION['user']);
$smarty->display("main.html");
```

```
}else{
    echo "<script>alert('您不具备访问权限！'); window.location.href='index.html';</script>";
}
?>
```

（4）创建模板页 main.html。首先，输出 PHP 动态页中定义的模板变量值，包括页面的标题（$title）、当前时间（$dates）和当前页输出的模块类别（$type、$caption）。然后，通过 include 函数加载模板变量 $admin_phtml 传递的模板页。最后，为后台管理系统中每个功能模块创建热点链接，并通过超链接的参数传递数据，同时应用 Smarty 模板中的 escape 方法对传递的参数值进行编码。关键代码如下：

```
<!--载入模板文件-->
{include file=$admin_phtml}
<!--创建热点链接-->
<map name="Map" id="Map">
<area shape="rect" coords="29,41,88,62" href="main.php?caption={"商品添加"|escape:"url"}&amp;type={"商品管理"|escape:"url"}" />
<area shape="rect" coords="30,71,91,90" href="main.php?caption={"商品修改"|escape:"url"}&amp;type={"商品管理"|escape:"url"}" />
<area shape="rect" coords="31,99,91,118" href="main.php?caption={"商品删除"|escape:"url"}&amp;type={"商品管理"|escape:"url"}" />
</map>
```

（5）创建热点链接中链接的动态 PHP 文件和模板文件。

DIY：改变上述实例中 SESSION 变量值在模板页中的输出方式。提示：通过{$smarty.session.user}输出 SESSION 变量 user 的值。（20 分）（实例位置：光盘\mr\17\qjyy\02_diy）

17.5.3　情景应用 3——html_option 函数向下拉列表中添加列表项

视频讲解：光盘\mr\17\lx\html_option 函数向下拉列表中添加列表项.exe
实例位置：光盘\mr\17\qjyy\03

在开发程序的过程中，都使用过下拉列表框，而且都知道下拉列表框的值是通过 option 标签定义的，可以直接在页面中定义值，也可以使用从数据库中读取的数据来定义，无论使用哪种都必须应用到 option 标签，这是在常规的 Web 页面中为下拉列表赋值的方法。

本实例中将介绍通过 Smarty 模板中的自定义函数 html_options 为下拉列表框赋值，使用 Smarty 模板中的自定义函数为下拉列表框赋值与使用 option 标签实现的效果是相同的，不过通过这种方法更加适合从数据库中读取数据，更符合 Smarty 开发模式动静分离的原则，运行结果如图 17.15 所示。



图 17.15　通过 html_options()函数为下拉列表赋值

自定义函数 html_options 根据给定的数据创建选项组。该函数可以指定哪些元素被选定，要么必须指定 values 和 ouput 属性，要么指定 options 替代。自定义函数 html_options 的语法如下：

```
<select name=customer_id>
    {html_options values=$cust_ids selected=$customer_id output=$cust_names}
</select>
```

html_options()函数的参数说明如表 17.6 所示。

表 17.6　html_options()函数的参数说明

参　　数	类　　型	说　　明
name	string	下拉菜单名称，默认值为空
options	associative array	包含值和显示的关联数组
selected	string/array	已选定的元素或元素数组
output	array	包含下拉列表各元素显示值的数组
values	array	包含下拉列表各元素值的数组

如果给定值是数组，将作为 OPTGROUP 处理，且支持递归。所有的输出与 XHTML 兼容。

如果指定可选属性 name，该选项列表将被置于<select name="groupname"></select>标签中，如果没有指定，那么只产生选项列表。

具体步骤如下：

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录；创建类文件 system.class.inc.php，定义数据库连接、操作类。创建 system.smarty.inc.php 文件，封装 Smarty 的配置类，定义类的实例化文件 system.inc.php，完成各个类的实例化操作，并返回操作对象。

（2）创建 index.php 文件。首先，设置页面的编码格式、载入配置文件。然后，生成随机验证码，并且将随机验证码的值赋给模板变量。接着，从数据库中读取数据，将读取的数据存储到数组中，并且将数组赋给模板变量，作为下拉列表中的值。最后，指定模板页。代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                 //设置文件编码格式
require_once("system/system.inc.php");                               //包含配置文件
$array= explode(' ', mt_rand(1000,9999));                            //生成随机验证码
$smarty->assign('title','html_option 函数向下拉列表中添加列表项');    //将指定数据赋给模板变量
$smarty->assign('content',$array);                                   //将数组赋给模板变量
$sql="select * from tb_bccd ";                                       //执行 select 查询语句
$res=$admindb->ExecSQL($sql,$conn);                                  //执行 select 查询语句
$array_id=array();
$array_name=array();
for($i=0;$i<count($res);$i++){                                       //循环读取二维数组中的数据
    $array_id[]=$res[$i]['id'];
    $array_name[]=$res[$i]['name'];
}
$smarty->assign('cust_id', $array_id);                               //设置下拉列表的值
$smarty->assign('cust_name', $array_name);                           //设置列表的显示数据
$smarty->display('index.html');                                      //指定模板页
?>
```

（3）创建 index.html 模板页。首先，创建 form 表单，添加表单元素，将用户的注册信息提交到 index_ok.php 文件中，其中，通过 html_options 自定义函数输出下拉列表框的值，通过 foreach 语句输出验证码的



值。然后，应用 literal 标签在页面中直接嵌入 CSS 样式和 JavaScript 脚本对用户提交的信息进行验证。关键代码如下：

```
<form id="form" name="form" method="post" action="index_ok.php" onsubmit="return check_form(); ">
词典选择：<select name="customer" id="customer">{html_options output=$cust_name values=$cust_id }</select>
验证码：  <input type="hidden" id="checks" name="checks" value="{foreach key=key item=item from=$content}
{$item}{/foreach}" />
<input  name="check"  type="text"  id="check"  size="8"  />{foreach  key=key  item=item  from=$content}
{$item}{/foreach}
<input type="image" name="imageField3" src="images/reg_07.jpg" />
<input type="image" name="imageField4" onclick="form.reset();return false;" src="images/reg_09.jpg" />
</form>
```

（4）创建 index_ok.php 文件，获取表单中提交的数据，将数据添加到数据表中。代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once("system/system.inc.php");                                  //包含配置文件
if($_POST['user']!=""&&$_POST['pass']!=""&&$_POST['checks']!=""){       //判断用户名和密码是否为空
        $sql="insert into tb_user(user,pass,dates,email,type)values('".$_POST['user']."','".md5 ($_POST['pass'])."',
'".date("Y-m-d")."','".$_POST['email']."','".$_POST['customer']."')";
        $res=$admindb->ExecSQL($sql,$conn);   //执行 insert 添加语句
        if($res){
            echo "<script>alert('用户注册成功！ '); window.location.href='main.php';</script>";
        }else{
            echo "<script>alert('用户注册失败！ '); window.location.href='index.php';</script>";
        }
}else{
        echo "<script>alert('用户注册信息不可为空！ '); window.location.href='index.php';</script>";
}
?>
```

（5）创建 main.php 和 main.html 文件，定义用户注册成功后跳转的页面。

**DIY：在 Smarty 模板中控制输出字符串的行宽。提示：应用{$content|wordwrap:30:"
\n"}。（20 分）（实例位置：光盘\mr\17\qjyy\03_diy）**

17.5.4　情景应用 4——Smarty 模板中应用正则表达式

视频讲解：光盘\mr\17\lx\Smarty 模板中应用正则表达式.exe

实例位置：光盘\mr\17\qjyy\04

本实例讲解 Smarty 模板中正则表达式的应用，通过该方法在 Smarty 模板页中完成字符串的替换操作，将输出的会员名称“mr”替换为“明日科技”，运行结果如图 17.16 所示。

regex_replace 变量是 Smarty 模板中的正则表达式，依据正则表达式对指定的字符串进行匹配。其包括两个参数：第一个是指定的正则表达式，第二个是指定的替换文本。其应用示例如下：

```
{$name|regex_replace:"/[mr]/":"<font color='#FF0000'>明日科技</font>"}
```

在这个示例中，将模板变量$name 中的“mr”替换为“明日科技”。

具体步骤如下：

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录；创建类文件 system.class.inc.php，定义数据库连接、操作类。创建 system.smarty.inc.php 文件，封装 Smarty 的配置类，定义类的实例化

文件 system.inc.php，完成各个类的实例化操作，并返回操作对象。



图 17.16　Smarty 模板中正则表达式的运用

（2）创建 index.php 文件。首先，通过 header()函数设置页面的编码格式，初始化 SESSION 变量。然后，根据 SESSION 变量判断当前用户是否具有访问权限。接着，如果具有访问权限，则应用 swicth 语句，根据超链接传递的参数值，完成在不同页面之间的跳转操作，即通过 include 语句包含不同的动态 PHP 文件，同时将动态 PHP 文件对应的模板文件名称赋给模板变量。最后，指定模板页 main.html。

（3）创建模板页 main.html。首先，输出 PHP 动态页中定义的模板变量值，包括页面的标题（$title）、当前时间（$dates）和当前页输出的模块类别（$type、$caption）。然后，通过 include 函数加载模板变量 $admin_phtml 传递的模板页。最后，为后台管理系统中每个功能模块创建热点链接，并通过超链接的参数传递数据，同时应用 Smarty 模板中的 escape 方法对传递的参数值进行编码。

（4）编辑 vip_look.php 文件，分页读取数据库中存储的会员数据，并且将分页读取返回的变量赋给模板变量，最终指定模板页。关键代码如下：

```
<?php
require_once("system/system.inc.php");    //调用指定的文件
//执行查询语句，从数据库中读取商品信息
$info=$admindb->ExecSQL("select count(*) as total1 from tb_name ",$conn);
$total1=$info[0]['total1'];                        //统计数据库中数据的总数
if(empty($_GET['pages'])==true || is_numeric($_GET['pages'])==false){     //判断变量 pages 是否为空
       $page1=1;                                   //如果变量为空，则赋值为 1
}else{                                             //如果不为空，则获取变量的值
       $page1=intval($_GET['pages']);
}
       $pagesize1=1;                               //定义每页显示 3 条记录
       if($total1<$pagesize1){                     //判断如果数据库中的数据小于每页显示的记录数
              $pagecount1=1;                       //则定义 pagecount 变量的值为 1
       }else{
              if($total1%$pagesize1==0){
                     $pagecount1=intval($total1/$pagesize1);  //用总的记录数除以每页显示的记录数，获取共有几页
              }else{
                     $pagecount1=intval($total1/$pagesize1)+1;
              }
       }
//将要输出的数据赋给 assign 模板变量
$smarty->assign("total1",$total1);
```

```
$smarty->assign("pagesize1",$pagesize1);
$smarty->assign("page1",$page1);
$smarty->assign("pagecount1",$pagecount1);
$array=$admindb->ExecSQL("select * from tb_name order by id desc limit ".($page1-1)*$pagesize1.",$pagesize1",
$conn );
if(!$array){
        $smarty->assign("iscommo","F");          //判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
        $smarty->assign("iscommo","T");          //判断如果执行成功，则输出模板变量 iscommo 的值为 T
       $smarty->assign("arr",$array);            //定义模板变量 arraybbstell，输出数据库中的数据
}
$smarty->assign(title,"会员浏览");
?>
```

（5）编辑 vip_look.html 模板页，通过 section 语句完成会员信息的分页输出，并且应用 regex_replace 变量将会员名称中的“mr”替换为“明日科技”。关键代码如下：

```
{section name=id loop=$arr}
   <tr>
      <td align="center" bgcolor="#FFFFFF">{$arr[id].id}</td>
      <td align="left" bgcolor="#FFFFFF">{$arr[id].name|regex_replace:"/[mr]/":"<font color='#FF0000'>明日科技
</font>"}</td>
      <td align="left" bgcolor="#FFFFFF">{$arr[id].email}</td>
      <td align="left" bgcolor="#FFFFFF">{$arr[id].dates}</td>
   </tr>
   {/section}
```

DIY：通过 Smarty 模板中的 escape 编码对超链接传递的参数进行编码处理。（20 分）（实例位置：光盘\mr\17\qjyy\04_diy）

17.5.5　情景应用 5——开启网站登录页面的缓存

视频讲解：光盘\mr\17\lx\开启网站登录页面的缓存.exe

实例位置：光盘\mr\17\qjyy\05

对于那些需要经常更新的程序，如果开启缓存，周期长短设置多长时间，这些都必须根据程序的实际需求进行设置。在本实例中开启网站登录页面的缓存，运行结果如图 17.17 所示。



图 17.17　后台登录

缓存的设置，既要考虑网站访问速度的提高，又要考虑缓存生命周期的合理性。不能偏执一方，应该采取中庸之道，达到一个最理想的效果。例如，网站首页是不需要经常更新的内容，缓存的生存周期就可以设置得长一些，而类似于论坛中帖子的数据，则可以不开启缓存，因为帖子的数据会不断地更新，就没有必要再使用缓存，如果使用缓存则可能会导致浏览不到最新的数据。

这里在 17.5.2 小节的基础上，开启用户登录页面的缓存。有关缓存的设置在 index.php 动态页中完成，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
require_once("system/system.inc.php");                        //包含配置文件
```

```
$smarty->caching=true;                                   //开启缓存
$smarty->cache_lifetime=3600;                            //设置缓存过期时间
if(!$smarty->is_cached('index.html')){
    $dates=date("Y-m-d H:i:s");
    $smarty->assign('lg_date', $dates);                  //设置下拉列表的值
}
$smarty->assign('title','开启网站注册页面的缓存');          //将指定数据赋给模板变量
$smarty->display('index.html');                          //指定模板页
?>
```

缓存文件存储在 system\Smarty\cache 目录下，而存储路径的设置是在 system\system.smarty.inc.php 文件中完成的。有关 Smarty 模板制作后台管理系统主页的其他功能，请读者参考 17.5.2 小节，这里不再赘述。

DIY：根据上述实例，在 index.php 文件中设置直接清除缓存的功能。（20 分）（实例位置：光盘\mr\17\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

17.6 自 我 测 试

一、选择题（每题 10 分，5 道题）

1．对 Smarty 模板描述错误的是（　　）。

A．Smarty 模板是 Zend 完全官方化的模板引擎

B．Smarty 模板具有强大的表现逻辑功能

C．Smarty 模板的特点是只适用于 PHP 程序员

D．Smarty 模板具有模板编译、缓存等优良的特征

2．在安装 Smarty 模板时，创建存储 Smarty 模板的目录是（　　）。

A．templates　　B．templates_c　　C．configs　　D．cache

3．Smarty 模板引擎需要在 PHP 的应用程序逻辑和页面模板中配合使用，才能完全分离表现层和逻辑层。在 PHP 程序中以下哪个步骤执行了编译的过程，将模板转换成 PHP 脚本（　　）。

A．加载 Smarty 模板引擎类，建立 Smarty 对象

B．修改 Smarty 的默认行为

C．将程序中动态获取的变量，通过 Smarty 对象中的 assign()方法置入模板中

D．利用 Smarty 对象中的 display()方法将模板内容输出

4．下面（　　）不是正确地在 Smarty 中使用变量的方式（　　）。

A．{func var="test $foo test"}　　B．{func var="test $foo [0] test"}

C．{func var="test $foo[bar] test"}　　D．{func var="test $foo.bar test"}

5．对 Smarty 模板引擎来说，缓存是必不可少的，下面（　　）是和缓存控制无关的 Smarty 属性。

A．$smarty->caching　　B．$smarty->cache_dir

C．$smarty->cache_lifetime　　D．$smarty->is_cache



二、填空题（每题 10 分，5 道题）

1．模板文件存储在（　　）文件夹下。
2．开启缓存的代码为（　　　　　　　　　）。
3．显示模板文件需要使用函数（　　　　　）。
4．为模板变量赋值需要使用函数（　　　　　）。
5．以下代码是模板页的相关内容，在 PHP 文件中为此循环赋值的代码为（　　　　）。

```
{foreach from=$value item=key}
            <tr>
                <td>{$key[0] }</td><td>{ $key[1] }</td><td>{ $key[2] }</td><td>{ $key[3] }</td>
            </tr>
{/foreach}
```

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

17.7　行 动 指 南

开始日期：______年____月____日　　　　　　　　　　　　　　　　　　　　结束日期：______年____月____日

序号	内　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）利用 Smarty 编写留言板。	
		（2）利用 Smarty 输出 SQL Server 数据库数据。	
		（3）利用 Smarty 开启页面缓存并操作页面。	
		（4）利用 Smarty 编写简单的新闻发布系统。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		

续表

序号	内　　容	行 动 指 南
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

17.8　成功可以复制——征途巨人史玉柱

成就：中国最具传奇色彩的创业人物，旗下的巨人网络集团成功登录美国纽约证券交易所，总市值达到 42 亿美元，成为在美国发行规模最大的中国民营企业。

创始公司：巨人网络公司，2007 年纽约证券交易上市，造就亿万富翁 21 位。

财富巅峰：2008 年荣誉《福布斯》互联网富豪榜第 7 位，个人资产 24 亿美元。

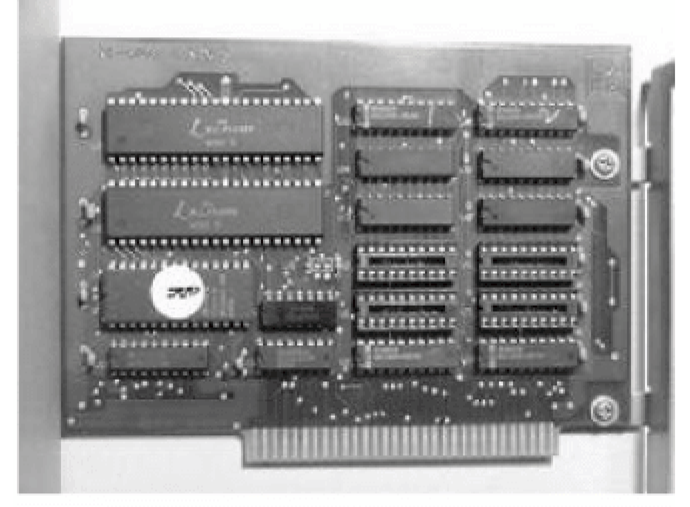
最高学历：硕士

出生日期：1962 年 9 月

超级语录：不要认为自己初中水平怎么样，初中水平跟博士后没啥区别。只要能干就行，我一直是这个观点，不在乎学历，只要能干能做出贡献就行。

成功历程

1980 年，史玉柱以优异的成绩考入浙江大学数学系。大学期间，史玉柱坚持每天长跑 18 公里，以此磨练意志和精神，这也为他日后做事百折不挠打下了坚实基础。1984 年毕业，史玉柱被包分配到了安徽省统计局农村抽样调查队。业余时间，史玉柱编写了一个程序，过去二三十个人忙活一年的活，一两天就干完了。搞得很多人从此没事干。干得起兴，史玉柱又编了一个分析软件，能分析出不同层次收入的农民可能会买什么东西，消费特征是什么，史玉柱的软件不断完善，以致得到了国家统计局的认可，要求全国各地的农村抽样调查都用史玉柱的软件。史玉柱因此得到了一二十元奖金和一个进步奖。相比当时每月 54 元的工资，史玉柱挺知足。


巨人汉卡图片

1988 年，史玉柱提交了辞职报告，在家潜心编写汉字文字处理软件。1989 年 7 月，史玉柱怀揣独立开发的汉卡软件和“M-6401 桌面排版印刷系统”软盘，南下深圳。当时，除了 4000 元钱，史玉柱一无所有。为了买到当时深圳最便宜的电脑（8500 元），他以加价 1000 元为条件，向电脑商获得推迟付款半个月的“优惠”，赊账得到了平生第一台电脑。为了推广产品，他用同样赊账的办法在《计算机世界》连续做了 3 期 1/4 版的广告。《计算机世界》给史玉柱的付款期限只有 15 天，可一直到广告见报后的第 12 天，史玉柱分文未进。就在关键时刻，第 13 天出现了转机：他一下子收到 3 张邮局汇款单，总金额 1.582 万元，先人一步的思维方式，让史玉柱迎来最初的成功。两个月后，他账上的金额竟达到了 10 万元。他再把钱投入广告中，扩大影响，4 个月后，仅靠卖 M-6401 产品就回款 100 万元，半年之后回款 400 万元。

1991 年 4 月，史玉柱带着汉卡软件和 100 多名员工来到珠海，注册成立珠海巨人新技术公司。但是刚刚把企业做大的史玉柱感受到了市场的巨大压力，其 M-6402 系列产品受到了来自香港金山电脑的强烈冲击。为了迅速打开市场，史玉柱又做了一次大胆的豪赌——向全国各地的电脑销售商发出邀请，只要订购 10 块巨人汉卡，史玉柱为他们报销路费，让他们前来珠海参加巨人汉卡的全国订货会，这种大胆的营销方式，快速建成了巨人汉卡的营销网络。1991 年，巨人汉卡的销量一跃成为全国同类产品之首，公司获纯利 1000 多万元。1992 年，巨人集团的资本超过 1 亿元，迎来第一个事业高峰，史玉柱也收获了人生的第一桶金。

深度评价

从巨人汉卡到巨人大厦，从脑白金到黄金搭档，史玉柱是具有传奇色彩的创业者之一。他曾经 5 年时间内跻身财富榜第 8 位；也曾一夜之间负债 2.5 亿；而如今他又是一个著名的东山再起者，保健巨鳄、网游新锐，身家数百亿的企业家。史玉柱的成功，突显出“执著与毅力”的魅力与价值。事业的跌宕起伏、世间的是非议论，唯有敢与苦难作伴的人，才能从跌倒的阴影中爬起来，迈向成功。

第18堂课

Zend Framework 框架

（视频讲解：145 分钟）

Zend Framework（简称 ZF）是由 Zend 公司支持开发的完全基于 PHP 5 的开源框架，采用 MVC 架构模式来分离应用程序中不同的部分，便于程序的开发和维护。拥有丰富的组件支持、模块化的结构设计、易于扩展和完善的文档资料以及灵活的架构设计。

学习摘要：

- Zend Framework 的 MVC 介绍
- Zend Framework 的 MVC 环境搭建
- Zend_Auth 身份认证
- Zend_Db 数据库操作
- Zend_Layout 网站布局
- Zend_Paginator 分页

18.1　Zend Framework 的 MVC 介绍

18.1.1　Zend Framework 概述

Zend Framework 框架是由 Zend 公司主创，Google、Microsoft 和 StrikeIron 作为合作伙伴为 Zend Framework 提供了大量技术和 Web 服务接口。

Zend Framework 中的组件都是独立的，都不依赖于其他组件，这样的松耦合结构可以让开发者独立使用组件。

Zend Framework 版本更新速度非常快，所包含的组件也在不断增加，所以说 Zend Framework 框架可以让一个从事 PHP 工作的人不掉队，始终保持在与最新技术的接触和学习当中。

Zend Framework 有自己成熟的社区，无论你有什么关于 Zend Framework 的问题，在社区里总能找到答案。

18.1.2　Zend Framework 常用组件

由于 Zend Framework 的组件都是非常独立的（这种情况称为松耦合），所以在开发 Web 程序或者软件时可以独立使用这些组件的组合。表 18.1 列举了 Zend Framework 的常用组件。

表 18.1　Zend Framework 的常用组件

组 件 名 称	功　　能
Zend_Acl	权限控制
Zend_Auth	主要用于认证
Zend_Cache	为应用程序提供缓存支持
Zend_Config	应用程序的配置数据参数
Zend_Db	提供 Zend Framework 与 MySQL 的数据库操作方式
Zend_File	开发者控制文件的上传和下载
Zend_Layout	实现应用程序的视图布局
Zend_Mail	实现 Zend Framework 发送和接收 E-mail
Zend_Paginator	提供数据的分页显示
Zend_Registry	提供对象注册表，相当于全局变量在程序中的使用
Zend_Session	提供 Zend Framework 的 session 控制
Zend_Validate	提供一组通用校验器

18.1.3　MVC 原理

MVC 是一种经典的程序设计理念，此模式将应用程序分为 3 个部分，即模型层（Model）、视图层（View）和控制层（Controller），MVC 是这 3 个部分英文字母的缩写。

MVC 设计模式产生的原因如下：

应用程序中用来完成任务的代码——模型层（也叫“业务逻辑”），通常是程序中相对稳定的部分，重



用率高。而与用户交互界面——视图层，却经常改变。

如果因需求变动而不得不对业务逻辑代码修改，或者因为要在不同的模块中应用到相同的功能而需要重复地编写业务逻辑代码，不仅降低整体程序开发的进度，也会使未来的维护变得非常困难。

因此将业务逻辑代码与外观分离，将会更方便地根据需求改进程序，这就是 MVC 设计模式。

在 PHP Web 开发中，MVC 设计模式的各自功能及相互关系如图 18.1 所示。

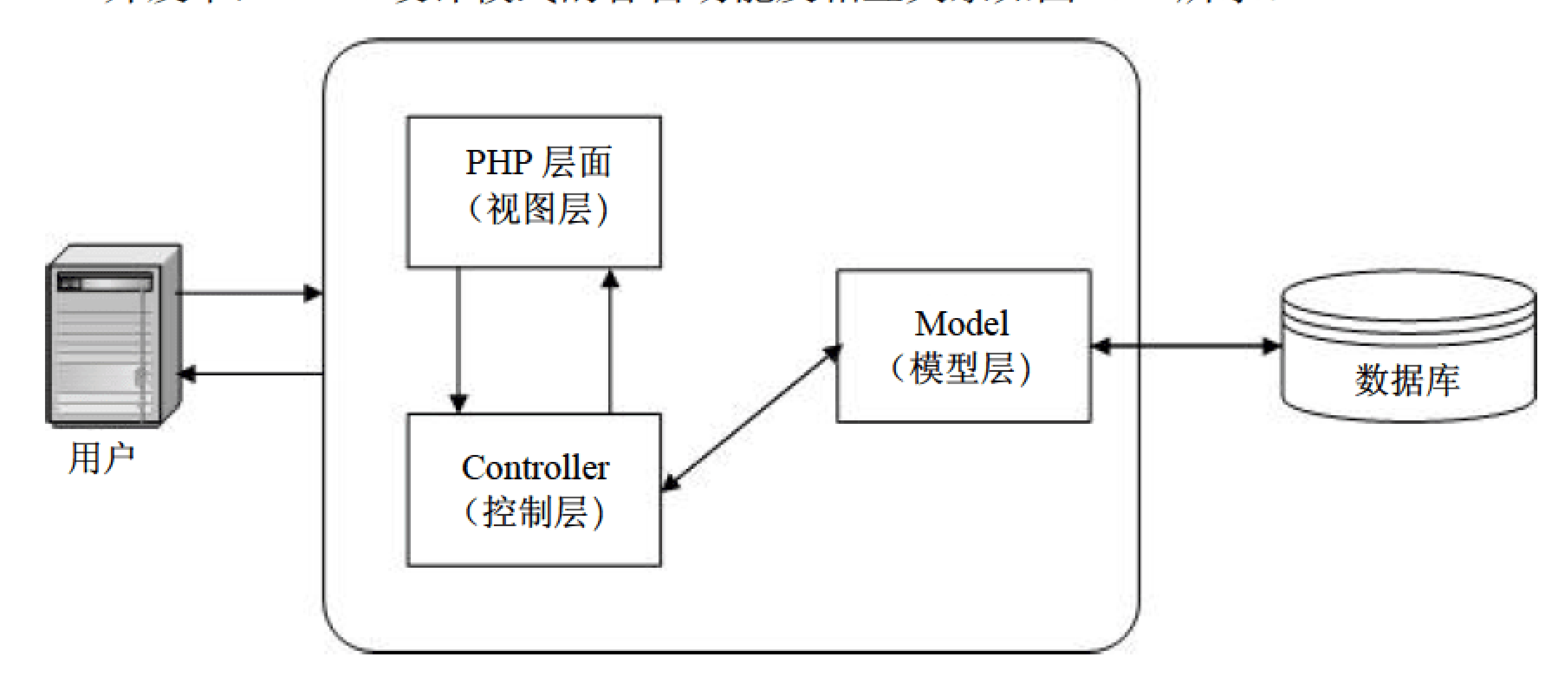


图 18.1　MVC 关系图

☑　模型层（Model）

模型层是应用程序的核心部分，它可以是一个实体对象或一种业务逻辑，它之所以称为模型，是因为它在应用程序中有更好的重用性和扩展性。

☑　视图层（View）

视图层提供应用程序与用户之间的交互界面，在 MVC 理论中，这一层并不包含任何的业务逻辑，仅提供一种与用户交互的视图。

☑　控制层（Controller）

控制层用于对程序中的请求进行控制，作用就像国家的宏观调控，它可以选择调用哪些视图或者调用哪些模型。

18.2　Zend Framework 的 MVC 环境搭建

Zend Framework 开发项目，首先必须保证 PHP 运行环境对 Zend Framework 框架的支持。Zend Framework 环境的搭建主要根据 PHP 环境而来，所以必须搭建 Apache 和 MySQL。

18.2.1　环境配置

搭建 Apache 和 MySQL，搭建的 Apache 中必须有 PDO_MYSQL 模块，如果安装环境中没有此模块可以去 PHP 官方网站中下载。

1．配置 httpd.Conf

（1）进入 Apache 的 conf 目录下，使用编辑工具打开 httpd.conf 文件，定位到如下位置：

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

（2）找到该文件中的“#”符号，该符号作用是在 conf 文件中注释掉该行。去掉“#”符号之后，表示加载 mod_rewrite 模块。

（3）在 Apache 加载 mod_rewrite.so 模块之后需要指定生效的目录，在 httpd.conf 文件中找到为“AllowOverride None”的所有字符串，将其修改为“AllowOverride All”，然后保存 httpd.conf 文件即可开启 mod_rewrite 功能。

技巧：在去掉 mod_rewrite.so 前面的“#”之后，需要从 hhtpd.conf 文件开始位置重新查找，才能保证将“AllowOverride None”完全查找。

2．配置 php.ini

Zend Framework 操作 MySQL 使用 PHP 自带的 PDO_MYSQL 模块。默认的 PHP 是不开启 PDO_MYSQL 模块的，所以必须重新对 PHP 环境进行配置，这里以 Windows 系统为例，在 C:\WINDOWS 下找到 php.ini 文件，定位到如下位置：

```
;extension=php_pdo.dll
......
;extension=php_pdo_mysql.dll
```

将上述两行代码前面的“;”（ini 文件的注释符）去掉，并保存 php.ini 文件。然后定位到 PHP 安装目录的 ext 文件夹下，在该文件夹中查看是否存在 php_pdo_mysql.dll 和 php_pdo.dll 文件，如果不存在这两个文件则需下载并存入 ext 文件夹中。至此 PDO_MYSQL 扩展加载成功。

最后重新启动 Apache 服务器，Zend Framework 运行环境配置成功。

提示：由于 Zend Framework 完全采用 PHP 5 的面向对象编程方式实现，所以 PHP 版本必须为 PHP 5 以上。同时为了发挥 Zend Framework 各个组件的功能，建议使用 PHP 5.2.6 及以上版本。如果是使用集成化安装包 AppServ 来搭建 PHP 开发环境，那么需要下载 AppServ 2.5.10 或者更高版本。

18.2.2 框架结构

要在项目开发中应用 Zend Framework 框架，只完成环境的配置是不够的，还要搭建一个完整的框架结构，才能在项目开发中发挥框架的作用。

自 Zend Framework 1.8 之后推出了 Zend_Application 组件使搭建结构更加方便和系统化。图 18.2 为一个相对完整的 Zend Framework 框架结构。

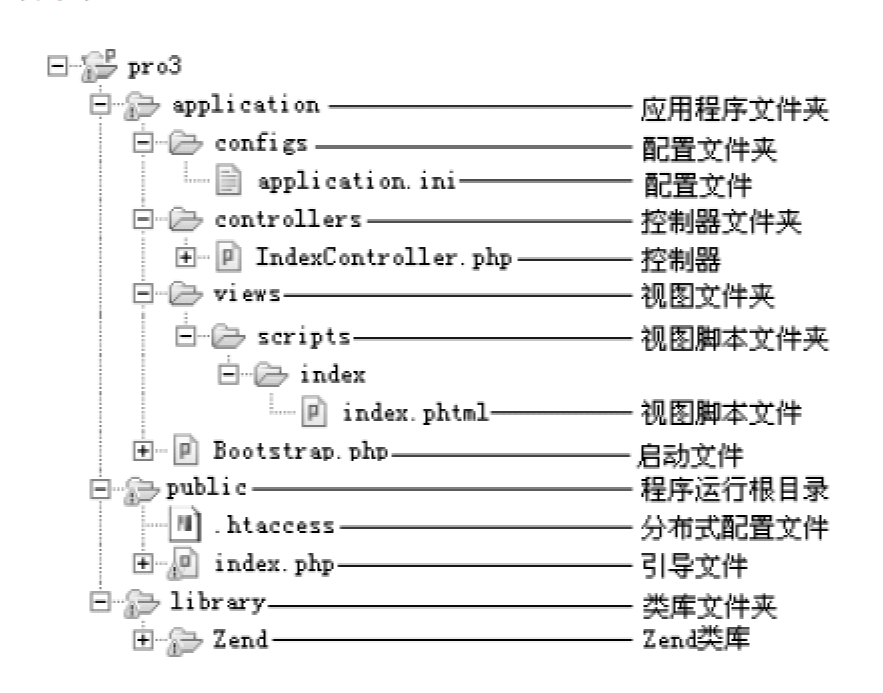


图 18.2　Zend Framework 框架结构



18.2.3　创建流程

了解框架的结构之后，下面讲解使用 Zend Framework 建立多模块 MVC 框架结构的具体流程。Zend Framework 框架的基本流程如图 18.3 所示。

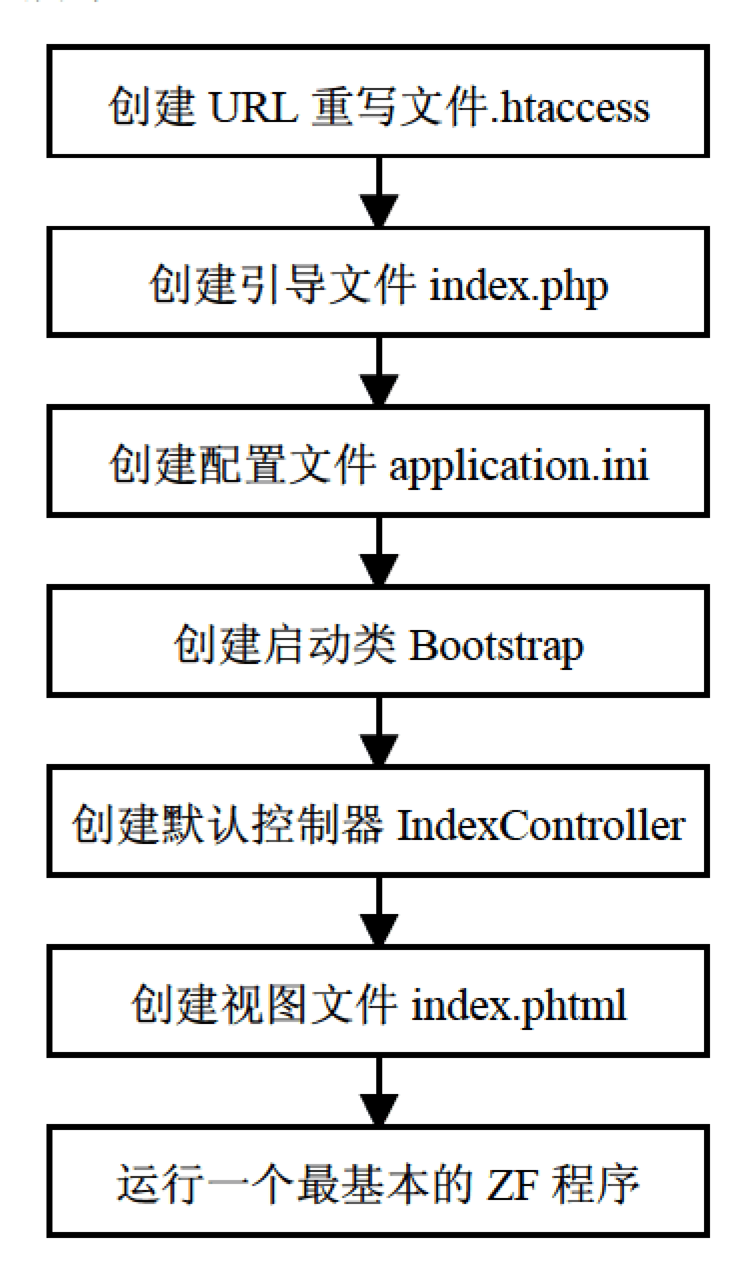


图 18.3　Zend Framework 框架的基本流程

例 18.01　开发一个简单的实例，输出“HELLO，ZF”，熟悉 Zend Framework 框架的基本流程。**（实例位置：光盘\mr\18\sl\18.01）**

具体步骤如下：

（1）在 public 目录下建立 URL 重写文件.htaccess，并在该文件中输入 URL 重写规则，如下所示。

```
#开启 URL 重写
RewriteEngine on
#除扩展名为.js、.css、.gif、.jpg、.png、.bmp 的文件外，访问其他文件都转向到 index.php 引导文件
RewriteRule !\.(js|css|gif|jpg|png|bmp)$ index.php
```

技巧： 由于.htaccess 文件没有文件主名，所以在 Windows 系统下无法直接命名，下面介绍两种创建.htaccess 文件的方法。

方法一：通过 Windows 系统的 copy con 命令创建，如图 18.4 所示。创建完成后使用 Ctrl+Z 组合键退出编辑模式。

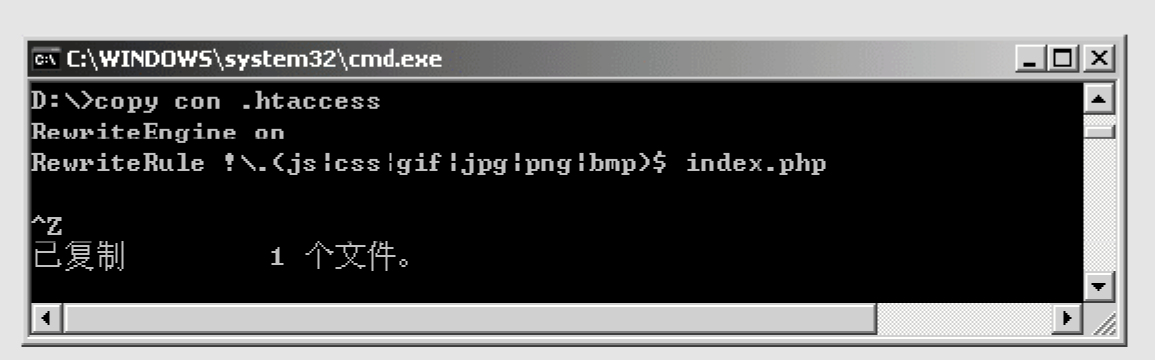


图 18.4　通过命令提示符创建.htaccess 文件

方法二：通过“记事本”文本编辑工具将文件另存为文件名为.htaccess 的文件，这里应该注意保存文件时需要选择文件类型为所有文件。

（2）在 public 目录下创建 index.php 引导文件，代码如下：

```
defined('APPLICATION_PATH') || define('APPLICATION_PATH', realpath(dirname(__FILE__) . '/../application'));
    //应用路径
defined('APPLICATION_ENV') || define('APPLICATION_ENV', getenv('APPLICATION_ENV') ? getenv('APPLICATION_
ENV') : 'project');                                                              //应用环境
$arrayIncludePath = array('.', realpath(dirname(__FILE__) . '/../../../library'));   //指定工程包含目录
set_include_path(implode(PATH_SEPARATOR, $arrayIncludePath));                   //将指定路径包含到工程中
require_once 'Zend/Application.php';                                            //包含 Application.php 文件
$application = new Zend_Application(APPLICATION_ENV, APPLICATION_PATH . '/configs/application.ini'); //
实例化 Zend_Application 类
$application->bootstrap()->run();                                               //调用启动文件并运行项目
```

上述代码中，首先定义两个常量 APPLICATION_PATH 和 APPLICATION_ENV，分别用来保存应用路径和应用环境名，这样在工程的其他模块中就可以直接使用。然后将当前目录地址“.”和 Zend Framework 类库地址作为数组元素保存在名为$arrayIncludePath 的数组中，并使用 implode()函数将上述路径用 PATH_SEPARATOR 常量连接，最后使用 set_include_path()方法将路径导入到工程中，这样工程就可以直接找到这些路径下的文件。

Zend Framework 类库被导入到工程后，就可以使用 require_once 等文件包含语句包含 Zend 目录下的 Application.php 文件，之后实例化 Zend_Application 类，并通过引用该类的 bootstrap()方法调用启动类，最终再通过调用 run()方法运行工程。

注意：在文件的结尾部分并没有加入“?>”，因为在文件的结尾它并不是必需的。这样就可以避免产生一些难于调控的错误问题。例如，在使用 header()函数来重新定向（redirect）时，如果在其前面某个包含文件中?>后面不小心加上了空格就会出现错误。

（3）在步骤（2）中实例化 Zend_Application 类时，为类的构造函数传入一个名为 application.ini 的文件，它是工程的配置文件。下面介绍 application.ini 文件的创建过程，其基本的内容如下：

```
 [project]
#设置错误级别
phpSettings.display_startup_errors = 1
phpSettings.display_errors = 1
#设置时区
phpSettings.date.timezone = 'Asia/Shanghai'
#配置启动类
bootstrap.path = APPLICATION_PATH "/Bootstrap.php"
bootstrap.class = "Bootstrap"
#配置工程模块
resources.frontController.moduleDirectory = APPLICATION_PATH "/modules"
```

配置文件存放位置和名称并不固定，只要与实例化 Zend_Application 类传递的参数相一致即可。

注意：这里配置的是多个工程模块，moduleDirectory=APPLICATION_PATH "/modules"，默认控制器放置在 defult 文件夹下，文件夹架构如图 18.5 所示。如果配置的是单个工程模块，controllerDirectory=APPLICATION_PATH "/controllers"，文件夹架构如图 18.2 所示。

（4）在 application.ini 配置文件中指定启动类的位置后，工程就可以找到启动文件 Bootstrap.php，关键



代码如下：

```
class Bootstrap extends Zend_Application_Bootstrap_Bootstrap{
//基本的启动类可以不进行任何操作
}
```

从上述代码可知，启动类 Bootstrap 继承自类 Zend_Application_Bootstrap_Bootstrap，该类中已经实现最基本的启动配置工作，所以在启动类中不需包含任何代码就可以运行最基本的 Zend Framework 应用。

（5）完成以上步骤后，就可以创建默认控制器 IndexController，同样一个最基本的控制器应该包含一个首页动作 indexAction。代码如下：

```
class IndexController extends Zend_Controller_Action{
    public function indexAction (){                          //默认动作
        $this->view->testStr="Hello ZF!";                    //为视图变量赋值
    }
}
```

说明：使用 Zend Framework 时，控制器名称应该以 Controller 结束，动作名应以 Action 结束。视图文件夹和控制器文件夹必须在同一目录下，这里都存储在 application\modules\default 文件夹下。

（6）完成默认控制器及首页动作的创建后，就可以创建视图文件 index.phtml。为了查看效果，在视图文件中仅输出首页动作中指定的视图变量的值，代码如下：

```
echo $this->testStr;
```

（7）多模块 MVC 框架结构的创建流程介绍完毕，最后看一下其完整的文件夹架构，如图 18.5 所示。

（8）假设上述工程所在目录为 Apache 默认主目录的 MR/18/sl/18.01 子目录下，则可以通过如下 URL 进行访问：

☑ http://127.0.0.1/MR/18/sl/18.01/public/。
☑ http://localhost/ MR/18/sl/18.01/public/index。
☑ http://localhost/ MR/18/sl/18.01/public/index/index。

其运行结果如图 18.6 所示。

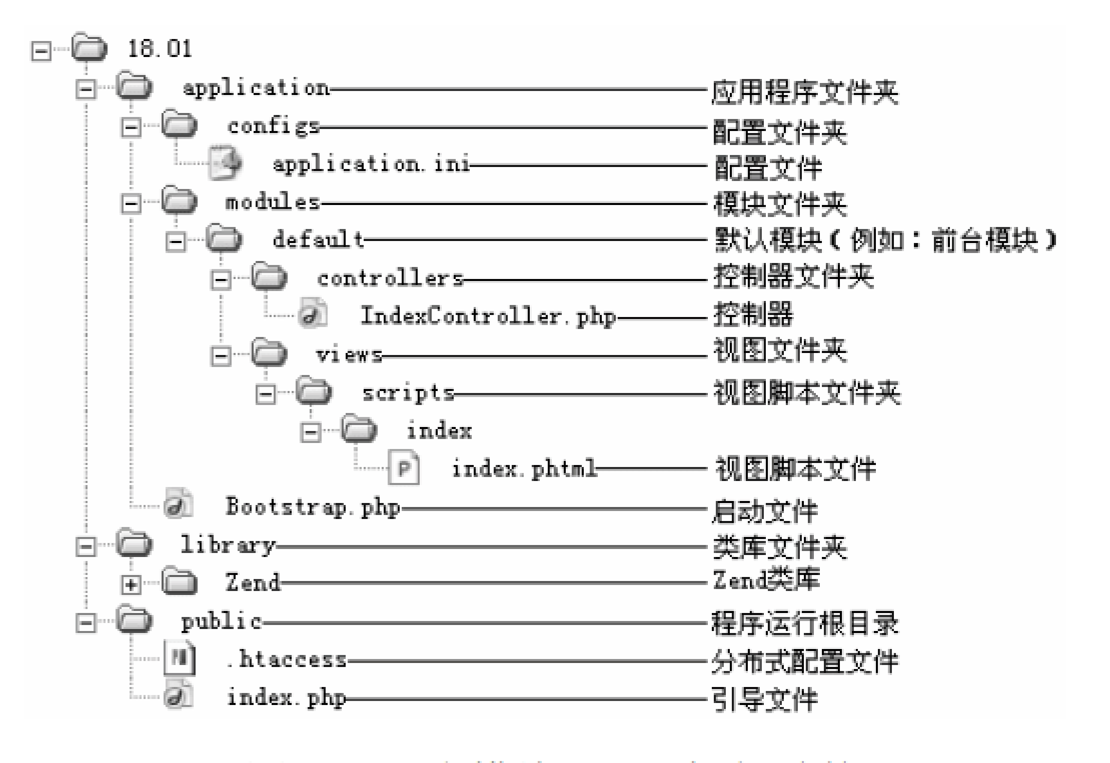


图 18.5　多模块 MVC 框架结构

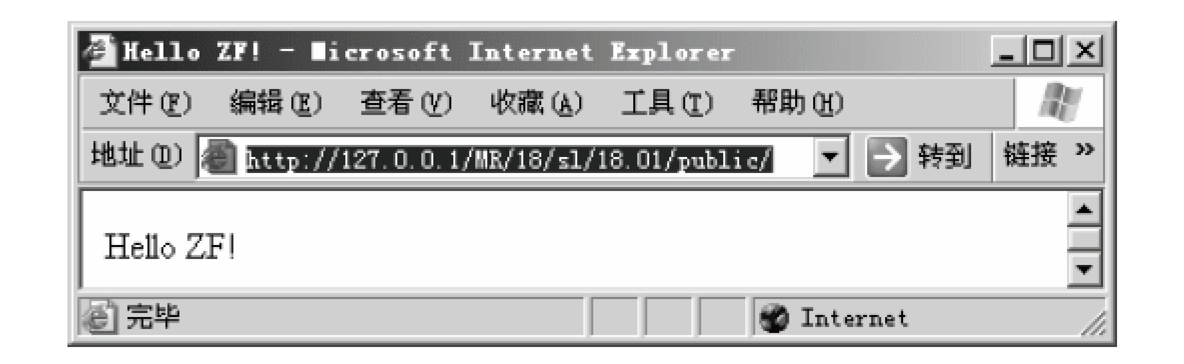


图 18.6　Zend Framework 框架创建演示

正常情况下，Zend Framework 的访问 URL 应该为“域名+模块名+控制器名+动作名”，那么为什么上述 URL 省略了部分内容也可以访问呢？这是因为在 Zend Framework 中，默认模块名、默认控制器名和首页动作名可以省略。

18.2.4 Zend Framework 的编码标准

对于程序人员必须养成良好的编码习惯，为未来打下一个良好的基础。在每一个项目中都会存在一种编码标准，这种编码标准会提高程序质量，减少 BUG，使维护和二次开发变得轻松。Zend Framework 编码标准是针对 Zend Framework 框架开发所设计的，它可以为参与 Zend Framework 开发的个人和团队所使用，也可以在没有特定编码标准的项目中使用。Zend Framework 编码标准介绍如下。

1．命名约定

☑ 类：类的名称只允许使用字母和数字，这里不鼓励使用数字，如果类名称中包含很多个单词，则每个单词的第一个字母必须大写。其中下划线只被允许分隔路径，如 Zend/Application.php 文件中对应的类名称应该为 Zend_Application。

☑ 文件名：文件名称只允许使用字母、数字、下划线和短横线（-），绝对禁止使用空格。同时任何存在 PHP 代码的文件都必须以.php 扩展名结尾（视图文件除外）。

☑ 函数和方法：函数名称只允许使用字母和数字，下划线是不允许使用的。这里同样不鼓励使用数字。函数的名称总是以小写开始，当包含多个单词时，每个单词的第一个字母大写。在对象方法中被定义为 private 或者 protected 的，名称必须以一个下划线开始。

☑ 变量：变量名称只允许使用字母和数字，下划线是不允许使用的。这里同样不鼓励使用数字。变量的名称总是以小写开始，当包含多个单词时，每个单词的第一个字母大写。在对象方法中被定义为 private 或者 protected 的，名称必须以一个下划线开始。

☑ 常量：常量名称只允许使用字母、数字和下划线。常量名称所有字母都必须为大写。每个单词都必须以下划线分隔。在对象方法中定义常量必须通过 const 定义类的成员，这里不鼓励使用 define 定义常量。

注意：除了类名称之外，其余命名都鼓励使用长字符，并能够说明函数的意图和行为。

2．编码风格

☑ 缩进：缩进是由 4 个空格组成，需要注意的是不允许使用 Tab 键。

☑ 行长度：一行代码的最佳长度在 80 字符以内。如果情况特殊可以延长到 120 个字符，但是绝对不能超过 120 个字符。

☑ 数组：当使用数组时每个逗号后都用一个空格来提高可读性。如果是关联数组则把代码分为多行，在每一个连续的开头用空格补充并对齐键和值。例如：

```
$Array = array( 'oneKey'    => 'oneValue',
                'twoKey' => 'twoValue');
```

3．注释文档

无论是在类，还是在包含 PHP 代码的文件中，必须至少在文件顶部包含一些注释信息，对类或者 PHP 代码进行解释说明。代码如下：

```
/**
*  文件/类的简短描述
*  文件/类的详细描述（如果有的话）… …
*/
```

函数和方法信息则必须在封装的函数上方给出注释信息。代码如下：



```
/**
* 函数的描述
* 所有参数
* 所有可能的返回值（类型）
* @return void
* 如果函数/方法抛出异常，使用@throws 于所有已经知道的异常类中
* @throws Zend_Application_Exception
*/
```

18.3　Zend_Auth 身份认证

Zend_Auth 是登录模块中对身份进行验证的组件，但并不是授权。Zend_Auth 具体作用是定义一些证书来确定身份是否是 Auth 声明的。其中这些“证书”叫做 Zend_Auth 适配器。

18.3.1　Zend_Auth 适配器

Zend_Auth 适配器是被用来认证特定的服务器。例如，接受认证证书，认证后返回的结果对于 Zend_Auth 适配器中都是通用的。身份认证必须符合程序要求，这就需要编写适合程序的适配器类。

Zend_Auth 适配器类必须实现 Zend_Auth_Adapter_Interface 类，调用接口函数 authenticate()，执行认证查询。

例如，简单介绍一个身份认证适配器类，仅针对固定用户名和密码，程序代码如下：

```
<?php
/**
 * 认证用户名和密码的适配器
 */
class Model_AuthAdapter implements Zend_Auth_Adapter_Interface{
    protected $_username;
    protected $_password;
    /**
     * Construct
     *
     * 带入用户名和密码
     *
     * @return void
     */
    public function __construct($username,$password){
        $this->_username = $username;
        $this->_password = $password;
    }
    /**
     * 对用户名和密码进行身份认证
     *
     * @throws Zend_Auth_Adapter_Exception If authentication cannot
     *                                      be performed
     * @return Zend_Auth_Result
     */
```

```
    public function authenticate(){
        $array = array();
        if (($this->_username == 'mr') && (($this->_password == 'mrsoft'))){
            $array[0] = true;
❶           return new Zend_Auth_Result(1,$array);
        }else{
            $array[0] = false;
            return new Zend_Auth_Result(-1,$array);
        }
    }
}
```

代码导读：

❶ Zend_Auth_Result 方法：将参数带入 Zend_Auth_Result 对象，验证时只需调用 Zend_Auth_Result 对象中的 isValid 方法。

Zend_Auth 适配器返回一个带有 authenticate()的 Zend_Auth_Result 的实例。在 Zend_Auth_Result 对象中，提供如下 4 种方法对 Zend_Auth 适配器返回结果进行操作。

- ☑ isValid()：返回 true 当且仅当结果表示一个成功的认证尝试。
- ☑ getCode()：返回一个 Zend_Auth_Result。常量标识符用来决定认证失败的类型或者是否认证成功。这个可以用于开发者希望区别若干认证结果类型的情形。
- ☑ getIdentity()：返回认证尝试的身份。
- ☑ getMessages()：返回认证尝试失败的数组。

18.3.2 身份持久认证

身份的持久认证必须结合 Zend_Session 组件，Zend_Session 组件通过会话（session）实现服务器端和客户端之间的持久联系。Zend_Session 采用 Zend_Session_Namespace 对象的方法管理。而且会话空间（Session Namespaces）提供了命名空间来管理会话数据。这样做的好处是可以在不同页面间调用不同的 Session 会话空间。

Zend_Session 主要的使用方法是：

```
<?php
include 'Zend/Session/Namespace.php';
$nameSpace = new Zend_Session_Namespace('Namespace');                    //开启 Session
```

下面通过实例讲解使用 Zend_Auth 和 Zend_Session 对身份持久认证。

例 18.02 Zend_Auth 和 Zend_Session 结合应用来对身份持久认证。（**实例位置：光盘\mr\18\sl\18.02**）

具体步骤如下：

（1）将 Zend_Auth 适配器类带入本实例中，放入 models 文件夹。根据 Zend Framework 编码标准重新编写适配器类名称，代码如下：

```
class Model_AuthAdapter implements Zend_Auth_Adapter_Interface{
    protected $_username;                               //声明成员变量
protected $_password;                                   //声明成员变量
public function __construct($username,$password){       //创建构造函数
        $this->_username = $username;                   //为成员变量赋值
        $this->_password = $password;                   //为成员变量赋值
}
```

```
public function authenticate(){                                   //定义方法对身份进行认证
        $array = array();
        if (($this->_username == 'mr') && (($this->_password == 'mrsoft'))){    //判断用户名和密码
            $array[0] = true;
            return new Zend_Auth_Result(1,$array);      //正确返回的结果
        }else{
            $array[0] = false;
            return new Zend_Auth_Result(-1,$array);     //错误返回的结果
        }
}
}
```

说明：在 Model 模型层写入的类可以直接在程序中实例化，而不需要引用文件。

（2）控制器中的功能包括验证用户名和密码、成功页面输出和安全退出。3 个功能分别对应 3 种方法。程序代码如下：

```
<?php
class IndexController extends Zend_Controller_Action{
public function indexAction(){
$this->view->assign("title","网站登录界面");
❶       if($this->_request->isPost()){                    //是否 POST 传输
$username = $this->_request->getPost('username'); //取得 username 值
$password = $this->_request->getPost('password');  //取得 password 值
$auth = Zend_Auth::getInstance();                   //调用 Zend_Auth
$authModel = new Model_AuthAdapter($username, $password);
//将适配器结果放入 Zend_Auth 的结果集中
❷           $result = $auth->authenticate($authModel);
if($result->isValid()){                             //使用默认验证
//开启 Session
$sessionNameSpace = new Zend_Session_Namespace('project');
$sessionNameSpace->username = $username;        //赋予 session 值
$sessionNameSpace->password = $password;
$this->_redirect('index/success');                 //跳转到成功页面
}else {
echo "用户名和密码错误";
}
}
}
public function successAction(){
$this->view->assign('title', '成功界面');
$sessionNameSpace = new Zend_Session_Namespace('project');
//判断 session 中是否设定 username 变量，即是否登录
❸       if($sessionNameSpace->username == ""){
die("不允许直接访问此页面");
}
$this->view->username = $sessionNameSpace->username;
$this->view->password = $sessionNameSpace->password;
}
❹   public function logoutAction(){
$sessionNameSpace = new Zend_Session_Namespace('project');
unset($sessionNameSpace->username);              //销毁 username 变量
```

```
$this->_redirect('index/index');
}
}
```

代码导读：

❶ **$this->_request->isPost()方法：判断当前页面是否使用 post 方式提交。**

❷ **authenticate 方法：适配器类中调用 Zend_Auth_Result 对象方法。**

❸ **加入不允许未登录访问本页面功能。通过 session 判断。**

❹ **安全退出方法：销毁 session 变量。**

（3）从控制器中可以看出存在 3 个页面，即 index、success 和 logout。但是仔细推敲后发现真正输出数据的页面只有 index 和 success。logout 页面没有任何数据输出，所以视图层中只建立 index.phtml 和 success.phtml 两个页面。index.phtml 页面的关键代码如下：

```
❶<title><?php echo $this->escape($this->title); ?></title>
<style type="text/css">
.submit {
	font-weight:bold;
	width:112px;
	border:medium none;
❷	background:#FFF url(<?php echo $this->baseUrl("images/index_02.jpg"); ?>) no-repeat;
	cursor:pointer;
	line-height:33px;
	font-size:14px;
}
</style>
❸<form id="form1" name="form1" method="post" action="<?php echo $this->Url(array('controller'=>'index',
'action'=>'index')); ?>">
<table width="338" height="168" background="<?php echo $this->baseUrl("images/index_01.jpg"); ?>">
    <tr>
        <td height="22"><input name="username" type="text" size="24" maxlength="10" /></td>
    </tr>
    <tr>
        <td><input name="password" type="password" size="25" maxlength="10" /></td>
    </tr>
</table>
<table width="338" height="62" background="<?php echo $this->baseUrl("images/index_04.jpg"); ?>">
    <tr>
        <td><input type="submit" name="submit" class="submit" value="   "/></td>
        <td><img src="<?php echo $this->baseUrl("images/index_03.jpg"); ?>" border="0"/></td>
    </tr>
</table>
</form>
```

代码导读：

❶ **$this->escape($this->title)：输出标题。**

❷ **$this->baseUrl("images/index_02.jpg")：设置背景颜色。**

❸ **$this->Url(array('controller'=>'index','action'=>'index'))：设置表单中数据提交的路径。**

 说明：在 form 表单中的 action 表示提交数据到哪一个程序。Zend Framework 框架中如果提交数据不是当前控制器必须使用$this->Url()方法确认控制器。

success.phtml 页面的代码如下：

```
<title><?php echo $this->escape($this->title); ?></title>
<table width="300" border="1" cellpadding="0" cellspacing="0" bordercolor="#333333">
  <tr height="30">
    <td width="60" align="center">用户名：</td>
    <td><?php echo $this->escape($this->username); ?></td>
  </tr>
  <tr height="30">
    <td align="center">密  码：</td>
    <td><?php echo $this->escape($this->password); ?></td>
  </tr>
  <tr height="30">
    <td><a href="<?php echo $this->Url(array('controller'=>'index','action'=>'logout')); ?>">安全退出</a></td>
  </tr>
</table>
```

技巧：在视图层中如果要调用图片或者各种文件，可以使用 Zend Framework 中的 baseUrl()函数调用，这样调用的位置就是入口文件所在文件夹。

运行结果如图 18.7 所示。



图 18.7　身份的持久认证

18.3.3　数据库认证

Zend Framework 框架中通过 Zend_Auth_Adapter_DbTable()组件实现数据库认证。

首先，应用 Zend_Db_Adapter_Abstract 连接一个数据库，并将返回对象传递给 Zend_Auth_Adapter_DbTable 的构造器。

然后，实例化 Zend_Auth_Adapter_DbTable，并通过配置选项传递参数，以及通过用户登录的用户名和密码，实现数据库的认证。

可用的配置选项介绍如下。

- ☑ tableName：包含认证证书的数据库表名，执行数据库认证查询需要依靠这个证书。
- ☑ identityColumn：数据库表的列名称，用来表示身份。身份列必须包含唯一的值，如用户名或者 E-mail 地址。
- ☑ credentialColumn：数据库表的列名称，用来表示证书。在一个简单的身份和密码认证 scheme 下，证书的值对应为密码。
- ☑ credentialTreatment：在许多情况下，密码和其他敏感数据是加密的。通过指定参数化的字串来使用这个方法，如'MD5(?)' 或者 'PASSWORD(?)'，开发者可以在输入证书数据时使用任意的 SQL 语句。

最后，调用 authenticate()方法执行认证查询。在认证结果对象中，可以通过 getIdentity()方法获取认证

的身份，也可以通过 getResultRowObject()方法返回一行数据。具体应用请参考基本功训练和情景应用。

18.4 Zend_Db 数据库操作

Web 语言编程离不开数据库的支持。Zend Framework 框架中连接和操作数据库是利用 Zend_Db 组件。Zend_Db 组件是一个基于 PDO 模块的数据库抽象层 api，它可以支持多种数据库，如 MySQL、SQLite、SQL Server 等。

18.4.1 Zend_Db_Adapter 数据库操作

应用 Zend_Db_Adapter 必须静态调用 Zend_Db::factory()方法。如调用 PHP 的最佳搭档 MySQL 数据库，程序代码如下：

```
<?php
include_once 'Zend/Db.php';                           //包含 Zend_Db 文件
$params = array(
        'host'=>        'localhost',                  //连接数据库 URL
        'username'      =>       'root',              //连接数据库用户名
        'password'=>         '111',                   //连接数据库密码
        'dbname'  =>         'db_database18',         //连接数据库的库名称
);
$db = Zend_Db::factory('PDO_MYSQL', $params);         //基于 PDO 连接 MySQL 数据库
```

以上就是一个简单的连接和操作数据库的方法。但是由于 Zend Framework 的版本升级到 1.8 以上，入口文件中并不需要调用此方法，所以此方法已经不能够满足现在的需求，当然并不是不可以使用。下面介绍的 Zend_Db_Table 是 Zend Framework 推荐的连接和操作数据库的组件。

18.4.2 Zend_Db_Table 数据库操作

Zend_Db_Table 是 Zend Framework 的表模块。它通过 Zend_Db_Adapter 连接到数据库，创建类来继承 Zend_Db_Table 类，实例化创建的类，通过返回的对象对数据表进行操作和查询。

例 18.03 在 Zend Framework 1.8 及以上版本中连接和操作 MySQL 数据库，查询表 tb_user 中的数据，并输出查询结果。**（实例位置：光盘\mr\18\sl\18.03）**

具体步骤如下：

（1）搭建 Zend Framework 的基本环境。

说明：从本节开始，将不再重复讲解 Zend Framework 框架的搭建。

（2）使用 MySQL 数据库，根据 Zend Framework 1.8 以上版本资源加载方法（详细情况查看 18.2.3 小节的基本环境搭建中启动文件），将连接数据库信息写入配置文件。程序代码如下：

```
resources.db.adapter = "PDO_MYSQL"                            # 基于 PDO_MYSQL 连接数据库
resources.db.params.host = "localhost"                        # 数据库 URL
resources.db.params.username = "root"                         # 数据库用户名
resources.db.params.password = "111"                          # 数据库密码
resources.db.params.dbname = "db_database18"                  # 数据库库名称
resources.db.params.driver_options.1002 = "set names utf8"    # 数据库字符集
```

（3）在启动文件中，通过_init 方法加载数据库资源，程序代码如下：

```
protected function _initDB(){
❶    $options = $this->getOption('resources');
❷    $options = $options['db'];
❸    $resources = $this->getPluginResource('db');
❹    $db = $resources->getDbAdapter();
     Zend_Db_Table::setDefaultAdapter($db);
     Zend_Registry::set('dbAdapter',$db);
     Zend_Registry::set('dbprefix',$options['params']['prefix']);
}
```

代码导读：

❶ getOption 方法：在配置文件中找到 resources 开头的资源配置。

❷ $options['db']：获取数据库资源，在资源配置中找到数据库（db）项。

❸ getPluginResource 方法：获取系统中的数据库资源。

❹ $resources->getDbAdapter()：定义数据库操作变量。在程序中只要通过调用 Zend_Registry::get 方法调用此变量，即可实现对数据库的操作。

（4）通过 Zend_Registry::get 方法调用数据库资源变量，通过 fetchAll 方法执行查询语句，最后通过 print_r 输出查询结果。程序代码如下：

```
public function indexAction(){
        //调用 db 对象 dbAdapter 为启动文件中设置调用资源
        $this->_db = Zend_Registry::get('dbAdapter');
        $sql = "select * from tb_user";                    //SQL 语句
        $result = $this->_db->fetchAll($sql);              //得到 SQL 语句结果，查看 fetchAll 方法
        print_r($result);
}
```

运行本实例，查看数据表中的信息，会抛出控制器错误，如图 18.8 所示。

```
Fatal error: Uncaught exception 'Zend_Controller_Dispatcher_Exception' with message 'Invalid
controller specified (error)' in F:\PkhPHP\www\MR\Instance\16\16.8
\library\Zend\Controller\Dispatcher\Standard.php:242 Stack trace: #0
F:\PkhPHP\www\MR\Instance\16\16.8\library\Zend\Controller\Front.php(946):
Zend_Controller_Dispatcher_Standard->dispatch(Object(Zend_Controller_Request_Http), Object
(Zend_Controller_Response_Http)) #1 F:\PkhPHP\www\MR\Instance\16\16.8
\library\Zend\Application\Bootstrap\Bootstrap.php(77): Zend_Controller_Front->dispatch() #2
F:\PkhPHP\www\MR\Instance\16\16.8\library\Zend\Application.php(358):
Zend_Application_Bootstrap_Bootstrap->run() #3 F:\PkhPHP\www\MR\Instance\16\16.8
\public\index.php(8): Zend_Application->run() #4 {main} thrown in
F:\PkhPHP\www\MR\Instance\16\16.8\library\Zend\Controller\Dispatcher\Standard.php on line 242
```

图 18.8　错误显示

这是因为 Zend Framework 框架找不到控制器（IndexController）对应的视图文件导致的，所以需要在视图文件夹（views）下创建一个 index.phtml 视图页面，在 index.phtml 页面中可以不加载任何程序。本实例的运行结果如图 18.9 所示。

```
Array ( [0] => Array ( [id] => 1 [login_name] => mr [login_pwd] => mrsoft )
        [1] => Array ( [id] => 4 [login_name] => mingri [login_pwd] => mrsoft )
        [2] => Array ( [id] => 5 [login_name] => king [login_pwd] => mrsoft )
        [3] => Array ( [id] => 6 [login_name] => king [login_pwd] => mrsoft ) )
```

图 18.9　运行结果

你问我答：通过 Zend_Db_Table 如何执行添加、删除和修改操作？

上述实例中应用的是 fetchAll()方法执行查询操作，返回查询结果。如果要执行数据的添加、删除和修

改则应用的是 insert()、delete()和 update()方法。
（1）添加数据：insert()方法，参数是列名:数据的关联数组。(Zend Framework)会自动对数据进行加引号处理，并返回插入的最后一行的 id 值（注意：这里不同于 zend_db_adapter::insert 方法，后者返回的是插入的行数）。
（2）删除数据：delete()方法，同时通过一个 where 条件语句决定需要删除的行。该方法将返回被删除的行数。(Zend Framework)不会对条件语句进行加引号处理，所以需要使用该表的 zend_db_adapter 对象调用 quoteInto()方法来定义条件语句。
（3）修改数据：update()方法，将列名:数据的关联数组作为参数，同时通过一个 where 条件语句来决定需要修改的行，该方法将返回被修改的行数。(Zend Frameword)将会自动对被修改的数据进行加引号处理，但是条件语句则需要使用该表的 zend_db_adapter 对象调用 quoteInto()方法来定义。

18.4.3　数据表类

通过继承 Zend_Db_Table 类，可以完成对数据表的操作，而对数据表的操作多数都是代码的重用，所以最好的方法就是将其定义到模型中，这样可以提高代码的重用率，同时也便于对程序的更新和维护。

通过模型类继承 Zend_Db_Table 类，封装对一个数据库的操作方法是非常简单的。数据表类中对数据库的操作主要分为 4 种，分别介绍如下。

☑ insert($dataArray)：添加数据，参数$dataArray 是对应数据表的联合数组。
☑ update($setArray,$where)：修改数据，参数$setArray 是对应数据表的联合数组，参数$where 是修改数据的条件，相当于 where 语句。
☑ delete($where)：删除数据，参数$where 是删除数据的条件，相当于 where 语句。
☑ fetchAll($where = null, $order = null, $count = null, $offset = null)：查询数据。如果是单纯的查询操作可以直接使用 fetchAll()方法，但是，当查询有很多附加条件时，那么就要使用 fetchAll()方法的参数。其参数说明如下。
  - $where：表示 where 语句。
  - $order：表示 order by 语句。
  - $count：表示 limit 语句中显示多少结果行。
  - $offset：表示 limit 语句中从第几个数据开始。

例 18.04　创建数据表类，通过数据表类完成对数据库的添加、删除、修改和查询。（**实例位置：光盘\mr\18\sl\18.04**）

具体步骤如下：

（1）在 public 目录下建立 URL 重写文件.htaccess，并在该文件中输入 URL 重写规则，代码如下：

```
#开启 URL 重写
RewriteEngine on
#除扩展名为.js、.css、.gif、.jpg、.png、.bmp 的文件外，访问其他文件都转向到 index.php 引导文件
RewriteRule !\.(js|css|gif|jpg|png|bmp)$ index.php
```

（2）在 public 目录下创建 index.php 引导文件，代码如下：

```
defined('APPLICATION_PATH') || define('APPLICATION_PATH', realpath(dirname(__FILE__) . '/../application'));
//应用路径
defined('APPLICATION_ENV') || define('APPLICATION_ENV', getenv('APPLICATION_ENV') ? getenv('APPLICATION_
ENV')   : 'project');                                                    //应用环境
$arrayIncludePath = array('.', realpath(dirname(__FILE__) . '/../../../library'));     //指定工程包含目录
```



```
set_include_path(implode(PATH_SEPARATOR, $arrayIncludePath));      //将指定路径包含到工程中
require_once 'Zend/Application.php';                                //包含 Application.php 文件
$application = new Zend_Application(APPLICATION_ENV, APPLICATION_PATH . '/configs/application.ini');
                                                                    //实例 Zend_Application 类
$application->bootstrap()->run();                                   //调用启动文件并运行项目
```

首先，定义常量 APPLICATION_PATH 和 APPLICATION_ENV，分别用来保存应用路径和应用环境名。然后，将当前目录地址“.”和 Zend Framework 类库地址作为数组元素保存在名为$arrayIncludePath 的数组中，并使用 implode()函数将上述路径用 PATH_SEPARATOR 常量连接。最后，使用 set_include_path()方法将路径导入到工程中，这样工程就可以直接找到这些路径下的文件。

Zend Framework 类库被导入到工程后，使用 require_once 包含语句包含 Zend 目录下的 Application.php 文件，之后实例化 Zend_Application 类，并通过引用该类的 bootstrap()方法调用启动类，通过调用 run()方法运行工程。

（3）在第（2）步实例化 Zend_Application 类时，为类的构造函数传入一个名为 application.ini 的文件，即工程的配置文件，其代码如下：

```
 [project]
#设置错误级别
phpSettings.display_startup_errors = 1
phpSettings.display_errors = 1
#设置时区
phpSettings.date.timezone = 'Asia/Shanghai'
#配置启动类
bootstrap.path = APPLICATION_PATH "/Bootstrap.php"
bootstrap.class = "Bootstrap"
#配置工程模块
resources.frontController.controllerDirectory = APPLICATION_PATH "/controllers"
#配置数据库 PDO 模块
resources.db.adapter = "PDO_MYSQL"
#设置数据库服务器
resources.db.params.host = "localhost"
#设置服务器用户名
resources.db.params.username = "root"
#设置服务器密码
resources.db.params.password = "111"
#设置数据库
resources.db.params.dbname = "db_database18"
#设置数据库编码格式
resources.db.params.driver_options.1002 = "set names utf8"
```

（4）在 application.ini 配置文件中指定启动类的位置后，工程就可以找到启动文件 Bootstrap.php，关键代码如下：

```
class Bootstrap extends Zend_Application_Bootstrap_Bootstrap{
public function _initAutoload(){
$moduleAutoloader = new Zend_Application_Module_Autoloader(array('namespace' => '','basePath' =>
'../application'));
return $moduleAutoloader;
}
}
```

启动类 Bootstrap 继承自类 Zend_Application_Bootstrap_Bootstrap，该类中已经实现最其本的启动配置

工作。

（5）创建模型 models 和数据表类 User，定义操作数据表的方法。代码如下：

```
<?php
class Model_User extends Zend_Db_Table{          //定义数据表类，继承 Zend_Db_Table
    protected $_name = "tb_user";                //定义数据表名称变量
    protected $_primary = "id";                  //定义数据 id 变量
    protected $_adapter;                         //定义数据库连接标识变量
    public function init(){                      //定义 init()方法
        $this->_adapter = $this->getAdapter();   //获取操作数据表的对象
    }
}
```

（6）创建控制器 controllers 和默认控制器 IndexController，定义添加、修改、删除和查询动作方法，实例化数据表（tb_user）模型中定义的 User 类，通过返回的对象调用数据库操作方法完成对数据库的操作。关键代码如下：

```
<?php
class IndexController extends Zend_Controller_Action{
    public function indexAction(){
        $table = new Model_User();                  //类的实例化
        //增加的数据
        $data = array(
            'login_name' => 'King',
            'login_pwd'  => 'mrsoft',
        );
❶       if($table->insert($data)){                  //执行添加操作
            $this->view->insert = "插入数据成功！";
        }
    }
    public function deleteAction(){                 //定义删除方法
        $table = new Model_User();                  //类的实例化
        $where = 'id = 2';                          //定义删除的条件
❷       if($table->delete($where)){                 //执行删除操作
            $this->view->delete = "删除成功！";
        }else {
            $this->view->delete = "该数据已经不存在";
        }
    }
    public function updateAction(){                 //定义更新方法
        $table = new Model_User();                  //类的实例化
        $set = array(
            'login_name' => 'mingri',
        );
        $where = 'id = 2';                          //定义更新条件
❸       if($table->update($set, $where)){           //执行更新操作
            $this->view->update = "修改成功！";
        }else {
            $this->view->update = "该数据不存在或已经被修改过";
        }
    }
    public function fetchAction(){                  //定义查询方法
```

```
            $table = new Model_User();                    //类的实例化
            //定义查询条件
            $where = null;
            $order = "id desc";                           //以 ID 降序排列
            $count = 3;                                   //输出 3 条记录
            $offset = 0;                                  //从第一条记录开始
❹           $result_all = $table->fetchAll($where, $order, $count, $offset);       //执行查询语句
            $this->view->select =$result_all;
      }
}
```

代码导读：

❶ insert()方法：相当于执行 insert into tb_user(login_name,login_pwd) values('king', 'mrsoft')语句。

❷ delete()方法：相当于执行 delete from tb_user where id = 2 语句。

❸ update()方法：相当于执行 update tb_user set login_name = 'mingri' where id = 2 语句。

❹ fetchAll()：可以无参数直接使用，但是当使用参数时，参数位置必须对应。

（7）控制器和动作方法创建完成后，创建与控制器对应的视图文件，这里创建 4 个视图文件，即 index.phtml、update.phtml、delete.phtml 和 fetch.phtml，输出对数据库的操作结果。在 fetch.phtml 视图文件中，通过 foreach 语句循环输出数据库的查询结果，其代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>查看数据</title>
<link href="<?php echo $this->baseUrl('css/style.css');?>" rel="stylesheet" type="text/css" />
</head>
<body>
<table width="600" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#336600">
      <tr>
            <td height="25" align="center" bgcolor="#FFFFFF">ID</td
            <td align="center" bgcolor="#FFFFFF">用户名</td>
            <td align="center" bgcolor="#FFFFFF">密码</td>
      </tr>
      <?php foreach ($this->select as $key => $select){?>
      <tr>
            <td height="20" align="center" bgcolor="#FFFFFF"><?php echo $select['id'];?></td>
            <td align="center" bgcolor="#FFFFFF"><?php echo $select['login_name'];?></td>
            <td align="center" bgcolor="#FFFFFF"><?php echo $select['login_pwd'];?></td>
      </tr>
      <?php }?>
      <tr>
            <td height="30" colspan="3" align="center" bgcolor="#FFFFFF">
            <table width="300" border="0" cellpadding="0" cellspacing="0">
                  <tr>
                        <td><a href="index">添加数据</a></td>
                        <td><a href="update">更新</a></td>
                        <td><a href="delete">删除</a></td>
                        <td height="28"><a href="fetch">查看</a></td>
                  </tr>
            </table>
```

```
            </td>
        </tr>
</table>
</body>
</html>
```

本实例的运行结果如图 18.10 所示。

ID	用户名	密码
58	King	mrsoft
57	King	mrsoft
56	King	mrsoft
添加数据　更新　删除　查看		

图 18.10　查看数据库中的数据

18.5　Zend_File 文件控制

Zend_File 组件主要用于实现文件的上传和下载，该组件设计上采用适配器方式，这样无论使用 HTTP 的 POST 方式还是 FTP 方式对文件进行上传，其具体实现代码都相同，如果上传方式发生改变，只需更改适配器即可。

由于该组件尚不成熟，目前只支持 HTTP 的 POST 方式上传，所以只能使用 Zend_File_Transfer_Adapter_Http 这个适配器实现文件的上传，但不久 Zend Framework 开发团队将推出更多的适配器（如支持 FTP 方式的适配器）。所以本节将主要介绍 Zend_File_Transfer_Adapter_Http 的使用方法。

18.5.1　使用 Zend_File_Transfer_Adapter_Http 实现 POST 方式文件上传

使用 Zend_File_Transfer_Adapter_Http 实现文件上传主要使用该类的 setDestination()和 receive()方法实现，其中 setDestination()方法用于指定上传文件保存在服务器端的目录；receive()方法用于执行上传操作。下面将通过一个简单的实例讲解使用该适配器类实现 POST 方式文件上传的方法。

例 18.05　使用 Zend_File_Transfer_Adapter_Http 适配器类实现文件上传。**（实例位置：光盘\mr\18\sl\18.05）**

为了能够掌握使用上述适配器类实现文件上传的核心方法，实现本实例时只建立一个文件上传表单和一个用于实现上传业务逻辑的控制器类。

具体步骤如下：

（1）建立 Zend Framework 的 MVC 框架结构，具体创建过程请参见 18.2 节，这里不再赘述。

（2）建立用于实现文件上传的表单。为了便于查看上传效果，在该表单中只设置用于选择上传文件的文件域和用于提交表单的图片按钮，其代码如下：

```
<form enctype="multipart/form-data" action="<?php echo $this->Url(array('controller'=>'index','action'=>'index')); ?>" method="POST">
    <span class="STYLE1">选择上传文件:</span>
<input name="uploadedfile" type="file" />  
<input type="image" name="imageField" src="<?php echo $this->baseUrl('images/sc.jpg');?>" />
</form>
```

通过 POST 方法实现文件上传的表单 form 标签中必须设置 enctype 属性的值为 multipart/form-data 方式，



并且 method 属性的值必须为 POST。

（3）建立上传表单后，就可以编写用于实现文件的业务逻辑。本实例实现文件上传的代码是在 IndexController 控制器类的 indexAction 动作中，其关键代码如下：

```
<?php
class IndexController extends Zend_Controller_Action{
    public function indexAction(){
        if($this->_request->isPost()){                          //判断是否是 POST 上传
            $adapter = new Zend_File_Transfer_Adapter_Http();
            $adapter->setDestination('./upfiles');              //设置上传文件存储路径
            if(!$adapter->receive()){                           //执行上传操作
                $messages = $adapter->getMessages();            //获取返回的错误信息
                echo implode("<br>",$messages);                 //输出错误信息
            }else{
                echo "<script>alert('上传成功！');</script>";
            }
        }
    }
}
```

在上述代码中，首先使用当前控制器对象的_request 属性的 isPost()方法判断是否已经提交了表单，如果是则进行文件上传操作。

在具体实现文件上传时，首先通过 new 关键字实例化 Zend_File_Transfer_Adapter_Http()适配器类，然后调用该适配器类的 setDestination()方法指定上传文件要保存的服务器端目录，最后通过调用该适配器类的 receive()方法实现文件上传。这里需要注意如果上传失败，则调用 receive()方法后将返回 FALSE 值，并可以通过上述适配器类的 getMessages()方法返回上传失败的原因。

运行上述实例，结果如图 18.11 所示，在表单的文件域中选择要上传的文件，然后单击“上传”图片按钮，则所上传的文件将被保存到该实例主目录的 upfiles 子目录中。



图 18.11　Zend_File_Transfer_Adapter_Http 实现 POST 方式文件上传

根据上述实例可知，通过 Zend_File_Transfer_Adapter_Http 适配器类实现文件上传的关键代码只有如下 3 行：

```
$adapter = new Zend_File_Transfer_Adapter_Http();           //实例化适配器类
$adapter->setDestination('./upfiles');                      //定义上传文件保存路径
$adapter->receive();                                        //执行上传操作
```

18.5.2　对上传文件的合理性验证

对文件上传时，有时需要对上传文件的类型、大小进行限制。例如，使用 PHP 开发的 Web 应用在制作文件上传模块中，为了提高项目的安全性，是不允许上传扩展名为.php 或.exe 的文件的，那么使用 Zend_File_Transfer_Adapter_Http 适配器类如何实现上传文件的有效性验证呢？

在具体讲解使用 Zend_File_Transfer_Adapter_Http 适配器类实现文件上传的合理性验证前，首先应该知

道常用的验证规则有哪些，其中表 18.2 为该适配器类常用的效验规则。

表 18.2　Zend_File_Transfer_Adapter_Http 适配器类常用的效验规则

效 验 规 则	说　　明
Count	指定上传文件的数量范围验证规则
ExcludeExtension	指定不允许的上传文件扩展名验证规则
Extension	指定允许的上传文件扩展名验证规则
ImageSize	指定上传图片大小的验证规则
IsImage	指定上传文件是否为图片的验证规则
size	指定上传文件大小的验证规则
ExcludeMimeType	指定上传文件不允许的 mime 类型验证规则
MimeType	指定上传文件允许的 mime 类型验证规则
Exists	指定要上传的文件已经存在于服务器的验证规则
NotExists	指定要上传的文件不存在于服务器的验证规则

了解表 18.2 中的常用验证规则后，又如何将这些规则应用于上传过程中呢？这里只需使用上传适配器类的 addValidator()方法在上传过程中指定上传规则即可，该方法一般包含 3 个参数，第一个参数用于指定表 18.2 中的验证规则名称；第二个参数为 boolean 型，如果为 TRUE 则当该条验证失败时继续向下验证，否则不再继续向下验证；第 3 个参数为数组型，用于设置具体验证方式和验证错误的提示信息。

例如，在图片上传时使用 Zend_File_Transfer_Adapter_Http 适配器类验证图片大小只能小于 1MB 的代码如下：

```
$adapter = new Zend_File_Transfer_Adapter_Http();                    //实例化适配器类
$adapter->setDestination('./upfiles');                               //定义上传文件保存路径
$adapter->addValidator('Size', false, array('min' => '0kB' , 'max' => '1MB' , 'bytestring' => false , 'messages' =>
'·您所上传的图片不能超过 1MB'));                                       //对文件大小进行验证
$adapter->receive();                                                 //执行上传操作
```

18.5.3　为上传增加过滤规则

通过适配器类的 addValidator()方法可以对上传文件进行有效验证，从而可以有效地提高系统的安全性和人性化程度，但在实际应用中，保存上传文件时需要更改上传文件名或对上传文件进行加密操作，这时就需要使用适配器类的 addFilter()方法为上传增加过滤规则，在具体讲解 addFilter()方法前，首先需要了解常用的过滤规则有哪些，其中表 18.3 为 Zend_File_Transfer_Adapter_Http 适配器类常用的过滤规则。

表 18.3　Zend_File_Transfer_Adapter_Http 适配器类常用的过滤规则

过 滤 规 则	说　　明
Decrypt	对上传文件进行解密过滤
Encrypt	对上传文件进行加密过滤
LowerCase	将上传的文本类型的文件中的英文字符转换为小写
UpperCase	将上传的文本类型的文件中的英文字符转换为大写
Rename	对上传文件进行重命名

使用 addFilter()方法为上传文件增加验证规则一般需要为该方法指定 3 个参数，第一个参数为要指定的过滤规则名称；第二个参数为数组型，用于为该过滤指定具体的过滤规则；第 3 个参数用来指定该过滤规

则要限定的文件域名称。

例如，在对图片上传时使用 Zend_File_Transfer_Adapter_Http 适配器类更改上传文件名为 newName.gif 的实现代码如下：

```
$adapter = new Zend_File_Transfer_Adapter_Http();                //实例化适配器类
$adapter->setDestination('./upfiles');                           //定义上传文件保存路径
$adapter->addFilter('Rename', array('target' => 'newName.gif', 'overwrite' => true), 'imagename');   //改文件名
$adapter->receive();                                             //执行上传操作
```

18.6 Zend_Layout 网站布局

通过 Zend_Layout 网站布局，可以提高网站页面的重用率，便于网站内容的更新和维护。例如，在一个网站中，每个页面中都可能包含相同的头文件和尾文件，所以在每个页面中都需要编写相同的头文件和尾文件。如果应用 Zend_Layout 对页面进行布局，将页面的头文件和尾文件定义到单独的模板中，这样在创建页面时就不用再编写相同的头文件和尾文件代码，只要调用指定的头和尾模板文件即可。

18.6.1 Zend_Layout 概述

在 Zend Framework 中，使用 Zend_Layout 组件实现对网站视图层页面的布局，从而可以有效地提高项目的可维护性、代码重用率和扩展能力。使用 Zend_Layout 对网站布局具有以下优点：

- ☑ 建立 Zend Framework 的 MVC 框架后，可以自动配置 Zend_Layout 的解析。
- ☑ 能够更好地实现业务逻辑和视图层的分离。
- ☑ Zend Framework 中允许在 application.ini 文件中配置布局名称、布局脚本及布局脚本路径。
- ☑ 在不需使用布局的动作中，可以通过代码取消布局。
- ☑ Zend_Layout 在没有建立 Zend Framework 的 MVC 框架的前提下可以独立使用。

18.6.2 Zend_Layout 使用方法

以往在使用 PHP 开发 Web 应用时，实现网站的布局是通过使用文件包含函数将网站头部文件、尾部文件及其他模块包含到程序中来完成的，而使用 Zend_Layout 实现网站的布局，只需简单的配置即可。下面以 Zend Framework 的 MVC 框架为例讲解利用 Zend_Layout 实现站点布局的方法和步骤。

（1）需要在 application.ini 文件中指定布局文件的保存目录和布局文件名，代码如下：

```
resources.layout.layoutPath = APPLICATION_PATH "/layouts"
resources.layout.layout = "default"
```

上述配置代码指定布局文件的保存目录为应用路径下的 layouts 目录，并指定该目录下的 default.phtml 文件为默认布局文件，这里需要注意在配置网站布局时，布局文件可以不指定，但需要在控制器的动作方法中通过如下代码指定：

```
$this->_helper->layout->setLayout('default');
```

（2）完成 application.ini 的布局配置后，就可以建立网站的头部导航（header.phtml）和尾部导航（footer.phtml）视图文件。为了能够通过 Zend_View 对象的 render()方法获取到这两个文件，需要将 header.phtml 和 footer.phtml 两个文件直接保存到框架的 scripts 目录下。

（3）建立完成网站头部导航和尾部导航后，就可以建立布局文件 default.phtml，根据步骤（1）中的配

置，default.phtml 文件被保存在应用路径的 layouts 目录下，关键代码如下：

```
<html>
    <head>
        <!--定网站头信息-->
    </head>
    <body>
        <!--包含头部导航-->
        <?php echo $this->render('default_header.phtml')?>
        <!--包含页面主体部分-->
        <?php echo $this->layout()->content;?>
        <!--包含尾部导航-->
        <?php echo $this->render('default_footer.phtml')?>
    </body>
</html>
```

从上述代码可知，在布局文件中包含网站的头部文件和尾部文件是通过视图对象的 render()方法实现的，包含网站的主体内容是通过视图对象的 layout()方法返回值的 content 属性实现的。

18.6.3　Zend_Layout 应用实例

为了能够更加清楚地掌握 Zend_Layout 组件实现网站布局的配置方法，下面将通过具体实例进行讲解。

例 18.06　使用 Zend_Layout 实现对明日 IT 新闻网的页面布局。**（实例位置：光盘\mr\18\sl\18.06）**

明日 IT 新闻网是一个专业的 IT 新闻类网站，在网站的首页、新闻列表页和新闻详细信息页面都使用相同的网站头部和尾部导航，为了能够提高代码的重用率及设计统一的页面风格，在实施网站布局时采用 Zend_Layout 组件实现。

具体步骤如下：

（1）建立 Zend Framework 的 MVC 框架结构，具体创建过程请参见 18.2 节，这里不再赘述。

（2）打开 application.ini 文件，加入如下配置代码来指定布局文件的保存目录。

```
resources.layout.layoutPath = APPLICATION_PATH "/layouts"
```

（3）建立明日 IT 新闻网的首部导航视图和尾部导航视图，其效果分别如图 18.12 和图 18.13 所示，并将这两个视图文件分别命名为 header.phtml 和 footer.phtml。



图 18.12　网站头部导航



图 18.13　网站尾部导航

（4）在网站布局目录下建立 default.phtml 视图文件，代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```



```
    <head>
        <?php
            echo $this->headMeta()->setHttpEquiv('content-type', 'text/html; charset=utf-8')
                                        ->appendHttpEquiv('x-ua-compatible', 'ie=7');
        ?>
        <?php echo $this->headLink()->setStylesheet($this->baseUrl('/css/style.css'))?>
        <style type="text/css">
            .default-body   {background:url(<?php   echo   $this->baseUrl('/img/bg.jpg')?>)   repeat-x   0   0;
background-color:#E7F6FB;}
            .default-top {width:980px; height:30px; clear:both;}
            .default-top .li1 {width:600px; height:30px; line-height:30px; text-align:left; color:#FFFFFF; float:left;}
            .default-top .li2 {width:200px; height:30px; line-height:30px; text-align:right; color:#FFFFFF; float:right;}
            .default-main {width:980px; background-color:#FFFFFF; clear:both;}
        </style>
    </head>
    <body class="default-body">
        <div class="default-top">
            <ul>
             </ul>
        </div>
        <div class="default-main">
            <?php echo $this->render('default_header.phtml')?>
            <?php echo $this->layout()->content;?>
            <?php echo $this->render('default_footer.phtml')?>
        </div>
        <div class="cell_h"></div>
    </body>
</html>
```

（5）建立 IndexController 控制器，并在其中建立 indexAction 动作，代码如下：

```
<?php
class IndexController extends Zend_Controller_Action{
    public function indexAction(){
         $this->_helper->layout->setLayout('default');      //指定布局文件为 default.phtml
     }
}
```

通过上面步骤即可实现通过 Zend_Layout 组件对网站进行布局，本实例的运行结果如图 18.14 所示。



图 18.14　明日 IT 新闻网网站布局效果

18.7　Zend_Paginator 分页

在制作网站的数据展示模块时，如果数据量较大，一般采用数据分页显示的方式，Zend_Framework 框架提供的 Zend_Paginator 组件是专门用于数据分页显示的，本节将对该组件进行介绍。

18.7.1　Zend_Paginator 简介

Zend_Paginator 是一款非常优秀的数据分页组件，该组件采用适配器方式，根据不同的适配器可以实现对数组、Zend_Select 查询结果的分页展示，该组件具有如下优点：

☑　对任意数据进行分页，而不是针对关系型数据库。

☑　针对需求数据进行呈现，而不是全部数据。

☑　不强制用户只使用一种途径呈现数据和渲染分页控件。

☑　可以单独使用（相对 Zend Framework 框架本身）。

18.7.2　Zend_Paginator 分页方法

使用 Zend_Paginator 组件实现数据的分页显示，可以简化操作，有效地提高程序的开发效率和后期维护工作。下面讲解使用 Zend_Paginator 实现数据分页的方法。

（1）使用 Zend_Paginator 组件实现数据分页，首先需要确定要使用的分页适配器，常用的分页适配器包括数组适配器和 Zend_Select 分页适配器，其中数组适配器使用 Zend_Paginator_Adapter_Array 类指定，Zend_Select 查询结果的分页显示所需的适配器使用 Zend_Paginator_Adapter_DbSelect 指定。例如，使用 Zend_Paginator_Adapter_Array 适配器的代码如下：

```
$array=array();                                        //定义要分页的数组
$adapter = new Zend_Paginator_Adapter_Array($array);  //实例化适配器
```

（2）创建完分页适配器后，即可对 Zend_Paginator 分页类进行实例化并指定分页参数，该过程的代码如下：

```
$paginator = new Zend_Paginator($adapter);             //实例化 Zend_Paginator 对象
$paginator->setPageRange(5)                            //指定分页控制器面板显示的页码个数
          ->setItemCountPerPage(20)                    //指定每页显示的记录数
          ->setCurrentPageNumber(1);                   //指定当前显示的页码
$this->view->paginator = $paginator;                   //将分页类赋值给视图变量
```

（3）获得 Zend_Paginator 分页类实例化对象后，就可以在视图层通过传递的分页类对象显示分页数据和分页控制面板，分页数据通过 foreach 语句循环输出。例如，通过如下代码显示分页数据中 ID 的值：

```
foreach($this->paginator as $data)                     //遍历分页类对象
{
    echo $date['id'];                                  //输出 ID 的值
}
```

（4）通过 foreach 语句循环输出指定页码的分页数据后，还需要制作用于实现分页控制的分页链接面板。制作分页链接控制面板的文件需要放置在框架的 scripts 目录中，该文件主要用于生成分页控制面板视图。分页链接控制面板文件的参数说明如表 18.4 所示。



表 18.4　分页链接控制面板文件的参数说明

参　数	说　明
first	首页页码
firstItemNumber	当前页码中第一条数据在完整数据集中的序号（位置）
firstPageInRange	程序运行开始时实现的页码
current	当前页码
currentItemCount	当前页数
itmCountPerPage	本页最多数据数量
last	最后页码
lastItemNumber	当前页码中最后一条数据在完整数据集中的序号（位置）
next	下一页页码
pageCount	总页数
pageInRange	现实页码数组
previous	上一页页码
totalItemCount	总数据数量

根据表 18.4 可知，在分页链接控制面板文件中可以通过如下代码显示“最后一页”导航链接。

```
<a href="<?php echo $this->baseUrl('/news/list?page='.$this->last)?>">最后一页</a>
```

（5）建立完分页链接控制面板页后，还需要在分页视图中使用视图对象的 paginationControl()方法导入分页面板控制页面。实现导入分页面板页面的代码如下：

```
<?php echo $this->paginationControl($this->paginator, 'Sliding', 'list_news_pagination_control.phtml', array('flag'
=>$this->flag, 'order'=>$this->order))?>
```

paginationControl()方法一般有 4 个参数，其中第一个参数用于指定分页控制面板关联的 Zend_Pageinator 对象；第二个参数用于指定控制面板样式，其中常用的控制面板样式说明如表 18.5 所示；第 3 个参数用来指定分页控制面板的页面文件名称；最后一个参数为数组型，用于向分页面板文件传递视图变量。

表 18.5　常用的控制面板样式说明

参　数	说　明
All	返回每个页码，这是下拉菜单跳转中的分页
Elastic	类似于 Google 分页样式
Jumping	跳转到最后一页之后，返回到首页页码
Sliding	类似于雅虎分页样式，以当前页为中心（推荐）

18.7.3　Zend_Paginator 分页应用

通过 18.7.2 小节的学习可以掌握使用 Zend_Paginator 实现数据分页的基本思路和方法，本节将通过具体实例讲解使用 Zend_Paginator 实现分页。

例 18.07　使用 Zend_Paginator 实现对明日 IT 新闻网的新闻主题的分页显示。**（实例位置：光盘\mr\18\sl\18.07）**

在制作明日 IT 新闻网的新闻主题展示页面时，使用 Zend_Paginator 组件实现主题的分页显示，下面具体讲解该模块的制作方法。

（1）建立 Zend Framework 的 MVC 框架结构，具体创建过程请参见 18.2 节，这里不再赘述。

（2）建立新闻表模型 Model_DbTable_News 类，并在该类中建立用于实现新闻分页的 findByPage 方法。application\models\DbTable News.php 文件的代码如下：

```
public function findByPage ($flag, $orderIndex, $page = 1, $itemCountPerPage = 10, $pageRange = 5)
{
    $select = $this->getAdapter()->select();                //生成 select 对象
    $select->from($this->_name);                            //指定要查询的表名
    if ($flag !== 'all') {                                  //指定查询条件
        if ($flag === 'ttj') {
        $where = 'istodaycommend = true';
        } else {
        $where = $this->getAdapter()->quoteInto('flag = ?', $flag);
        }
        $select->where($where);
    }
    if ($orderIndex == 0) {                                 //排序方式
        $order = 'addtime desc';
    } elseif ($orderIndex == 1) {
        $order = 'addtime';
    } elseif ($orderIndex == 2) {
        $order = 'browse desc';
    } elseif ($orderIndex == 3) {
        $order = 'browse';
    }
    $select->order($order);                                     //排列顺序
    $paginatorAdapter = new Zend_Paginator_Adapter_DbSelect($select); //实例化并构建 Zend_Paginator 对象
    $paginator = new Zend_Paginator($paginatorAdapter);
$paginator->setPageRange($pageRange)->setItemCountPerPage($itemCountPerPage)->setCurrentPageNum
ber($page);                                                     //指定分页参数
    return $paginator;                                          //返回 Zend_Paginator 对象
}
```

上述代码首先构建 Zend_Select 查询对象，然后使用 Zend_Paginator_Adapter_DbSelect 分页适配器生成用于对 Zend_Select 查询结果进行分页的适配器对象，同时实例化 Zend_Paginator 分页类对象并指定分页参数，最后返回分页类对象。

（3）在 IndexController 控制器的 indexAction 动作中，调用新闻表模型的 findByPage()方法，并将返回结果赋值给视图变量，以便在视图层应用。application\controllers\ IndexController.php 文件的代码如下：

```
public function indexAction ()
{
    //接收参数
    if($this->_request->getParam('flag')=="" || $this->_request->getParam('order')=="" ||$this->_request->getParam
('page') ==""){
        $flag ="all";
        $order = 2;
        $page = 1;
    }else{
        $flag = $this->_request->getParam('flag');
        $order = $this->_request->getParam('order');
        $page = $this->_request->getParam('page');
    }
    //分页参数
```

```
        $itemCountPerPage = 10;
        $pageRange = 5;
        //视图变量赋值
        $this->view->flag = $flag;
        $this->view->order = $order;
        //新闻分页
        $paginator = $this->_news->findByPage($flag, $order, $page, $itemCountPerPage, $pageRange);
        $this->view->paginator = $paginator;
        $this->_helper->layout->setLayout("default");
    }
```

（4）在视图层，使用 foreach 循环语句输出新闻主题信息。application\views\scripts\index\index.phtml 文件的代码如下：

```
<?php foreach ($this->paginator as $news):?>
    <div class="content">
        <div class="title">
            ·<a href="#" class="a25"><?php echo $this->escape($news['title'])?></a>
        </div>
        <div class="description">
            <?php echo trim($this->escape($this->substr($news['uncontent'], 220)))?> <a href="#" class=
"a3"><nobr>&lt;详细&gt;</nobr></a>
        </div>
        <div class="msg">
            来源：<font color="#888888"><?php echo $this->escape($news['source'])?></font> | 作者：<font
color="#888888"><?php echo $this->escape($news['author'])?></font> | 时间：<font color="#888888"><?php
echo substr($news['addtime'], 0, 16)?></font> | 浏 览 ： <font color="#888888"><?php echo
$news['browse']?></font> 次
        </div>
    </div>
<?php endforeach;?>
```

（5）在 index.phtml 页面中使用 paginatorControl()方法调用分页控制面板，代码如下：

```
<?php echo $this->paginationControl($this->paginator, 'Sliding', 'list_news_pagination_control.phtml', array('flag'
=>$this->flag, 'order'=>$this->order))?>
```

（6）建立分页控制面板。application\views\scripts\ list_news_pagination_control.phtml 文件的代码如下：

```
<div style="width:280px; height:20px; line-height:20px; color:#0257AE; font-weight:bold; text-align:center;
border:1px solid #92C7ED; background-color:#FAFEFF; margin-right:8px; float:left;">
每页 <font style="color:#999999; font-family:Arial; font-size:12px;"><?php echo $this->itemCountPerPage?>
</font> 条/共 <font style="color:#999999; font-family:Arial; font-size:12px;"><?php echo $this->
totalItemCount?></font> 条  第 <font style="color:#999999; font-family:Arial; font-size:12px;">
<?php echo $this->current?></font>  页 / 共  <font style="color:#999999; font-family:Arial;
font-size:12px;"><?php echo $this->pageCount?></font> 页
</div>
<?php if($this->totalItemCount > 0):?>
    <?php if($this->current > 1):?>
    <div style="width:40px; height:20px; line-height:20px; border:1px solid #92C7ED; background-color:<?php
if($this->current==1):?>#1A75D2;<?php else:?>#FAFEFF;<?php endif;?> font-weight:bold; float:left; margin-
right:8px;">
        <a href="<?php echo $this->baseUrl('/list-'.$this->flag.'-'.$this->order.'-1.html')?>" class="<?php if($this->
current==1):?>a4<?php else:?>a3<?php endif;?>">首页</a>
    </div>
    <?php endif;?>
```

```
    <?php foreach ($this->pagesInRange as $page):?>
    <div style="width:20px; height:20px; line-height:20px; border:1px solid #92C7ED; background-color:<?php if($this->current==$page):?>#1A75D2;<?php else:?>#FAFEFF;<?php endif;?> font-family:Arial; font-style:italic; font-weight:bold; float:left; margin-right:8px;">
        <a href="<?php echo $this->baseUrl('/list-'.$this->flag.'-'.$this->order.'-'.$page.'.html')?>" class="<?php if($this->current==$page):?>a4<?php else:?>a3<?php endif;?>"><?php echo $page?></a>
    </div>
    <?php endforeach;?>
    <?php if($this->pageCount > $this->pageRange && $this->pageCount!=$this->current):?>
    <div style="width:40px; height:20px; line-height:20px; border:1px solid #92C7ED; background-color:<?php if($this->current==$this->last):?>#1A75D2;<?php else:?>#FAFEFF;<?php endif;?> font-weight:bold; float:left; margin-right:8px;">
        <a href="<?php echo $this->baseUrl('/list-'.$this->flag.'-'.$this->order.'-'.$this->last.'.html')?>" class="<?php if($this->current==$this->last):?>a4<?php else:?>a3<?php endif;?>">尾页</a>
    </div>
    <?php endif;?>
<?php endif;?>
```

以上为明日 IT 新闻网新闻主题循环输出的实现过程，本实例的运行结果如图 18.15 所示。



图 18.15　明日 IT 新闻网新闻主题分页展示

18.8　照猫画虎——基本功训练

18.8.1　基本功训练 1——通过 Zend_Acl 控制角色和资源的权限

视频讲解：光盘\mr\18\lx\通过 Zend_Acl 控制角色和资源的权限.exe

实例位置：光盘\mr\18\zmhh\01

Zend_Acl 是通过访问列表的形式来实现权限管理，应用程序可以通过 Zend_Acl 轻松制定出限制某些特定对象（角色）访问特定受保护的对象（资源）。权限控制的主要结构有两个，即 resource（资源）和 role（角色）。例如，拥有钥匙的人进入一个房间，这时拥有钥匙就相当于 role，这个房间就是 resource，因为想要进入房间并不是所有的钥匙都可以的。

1．角色（role）

在 Zend_Acl 中创建 role 是非常方便的，程序员只需要在 Zend_Acl 中增加 Zend_Acl_Role 的实现类即可，



如增加名称为“guest”的 role。代码如下：

```
$acl = new Zend_Acl();
$acl->addRole(new Zend_Acl_Role('guest'));
```

说明：这是为了让 Zend_Acl 将一个对象当作一个 role，所以使用 addRole 增加 role。

Zend_Acl 中对于角色的规则中存在“继承”，就是说一个角色可以继承另外一个或多个角色。例如，增加名称为“guest”的角色，拥有另外一个角色的权限。代码如下：

```
$acl = new Zend_Acl();
$acl->addRole(new Zend_Acl_Role('member'))
    ->addRole(new Zend_Acl_Role('admin'));
$parents = array('member', 'admin');
$acl->addRole(new Zend_Acl_Role('guest'), $parents);
```

注意：角色继承规则中，一个角色继承另外多个角色（通过数组），如果多个角色之间存在冲突，那么角色首先继承的是最后列出的角色。

2. 资源（resource）

在 Zend_Acl 中创建 resource 同样也是非常方便的，程序员只需要在 Zend_Acl 中增加 Zend_Acl_Resource 的实现类即可，如增加名称为“muchResource”的 resource，代码如下：

```
$acl = new Zend_Acl();
$acl->add(new Zend_Acl_Resource('muchResource'));
```

Zend_Acl 中对于 resource 允许其简单的规则继承。如果一个 resource 从唯一的一个父 resource 继承，而这个父继承 resource 同样也可以有其父 resource。例如：

```
$acl = new Zend_Acl();
$acl->add(new Zend_Acl_Resource('granddad'));
$acl->add(new Zend_Acl_Resource('father'), 'granddad');
$acl->add(new Zend_Acl_Resource('son'), 'father');
```

Zend_Acl 可以利用角色和资源对权限进行管理，在本实例中，使用 isAllowed()方法查询角色是否拥有执行功能的许可。运行结果如图 18.16 所示。

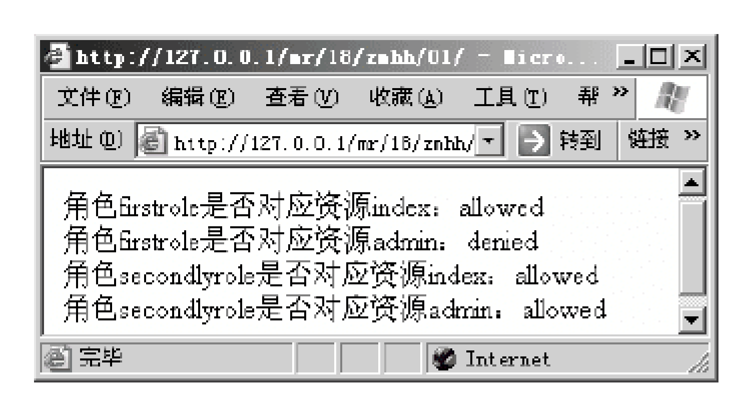


图 18.16 查询角色权限

在本实例中没有搭建 Zend Framework 框架，只是将 Zend 类库引入该实例中。创建 PHP 程序页面 index.php，代码如下：

```
<?php
header("Content-Type: text/html; charset=utf-8");
$arrayIncludePath = array('.', realpath(dirname(__FILE__) . '/../../../library'));    //指定工程包含目录
set_include_path(implode(PATH_SEPARATOR, $arrayIncludePath));                       //将指定路径包含到工程中
include('Zend/Acl.php');
include('Zend/Acl/Role.php');
include('Zend/Acl/Resource.php');                                                   //实例化 Zend_Acl
$acl = new Zend_Acl();
```

```
//增加 role
$acl->addRole(new Zend_Acl_Role('firstrole'));
$acl->addRole(new Zend_Acl_Role('secondlyrole'), 'firstrole');
//增加资源
$acl->add(new Zend_Acl_Resource('index'));
$acl->add(new Zend_Acl_Resource('admin'));
❶//赋予 role 到 resource
$acl->allow('firstrole', 'index');
$acl->allow('secondlyrole', 'admin');
❷//查看各个 role 是否对应到了 resource
echo "角色 firstrole 是否对应资源 index：" , $acl->isAllowed('firstrole', 'index') ? 'allowed' : 'denied';
echo "角色 firstrole 是否对应资源 admin：" , $acl->isAllowed('firstrole', 'admin') ? 'allowed' : 'denied';
echo "角色 secondlyrole 是否对应资源 index：" , $acl->isAllowed('secondlyrole', 'index') ? 'allowed' : 'denied';
echo "角色 secondlyrole 是否对应资源 admin：" , $acl->isAllowed('secondlyrole', 'admin') ? 'allowed' : 'denied';
```

代码导读：

❶ Zend_Acl 对角色赋值使用 allow()函数，也可以让角色不被赋予某个资源需使用 deny()函数。

❷ Zend_Acl 查询权限使用 isAllowed()函数，在本程序中使用的是三元运算。

照猫画虎：以上介绍了简单的 Zend_Acl 访问控制，但是在具体的应用程序中这样的访问控制肯定是不能够满足需求的，这时 Acl 也给出了精细的访问控制，通过 allow()函数的第 3 个参数来控制访问。（20 分）（实例位置：光盘\mr\18\zmhh\01_zmhh）

18.8.2 基本功训练 2——比较缓存日期与本地日期

视频讲解：光盘\mr\18\lx\比较缓存日期与本地日期.exe

实例位置：光盘\mr\18\zmhh\02

缓存是提高页面浏览速度的最佳方法，但并不是所有的页面或者数据都适合被缓存，缓存需要遵守一定的规则：

☑ 生成页面时，数据输出是相同的，如首页。

☑ 如果存在敏感信息的输入产生的页面不可以使用缓存，如注册页面。

☑ 信息更新非常频繁的数据和页面不可以使用缓存，如聊天室。

Zend_Cache 提供一个缓存任何数据的方法。在 Zend Framework 框架中将 Zend_Cache 缓存分为前端和后端，前端用来缓存操作，后端则设置各种缓存条件，进而做到缓存任何想要的页面或者数据，也可以删除特定的页面或者数据。例如，可以删除所有名称为 ids 的数据等。

Zend_Cache 的缓存原理是，缓存数据是程序在第一次执行时，把需要缓存的数据或者页面进行缓存处理，Zend_Cache 是将数据缓存到文件或者内存中，当以后的运行需要调用此数据时，可以直接从缓存中取出缓存结果而不需要再次运行此程序。

1. Zend_Cache 工厂方法

Zend_Cache 工厂方法用来构造 Zend_Cache 前端，该方法的语法如下：

```
$cache = new Zend_Cache:: factory($frontendName, $backendName,$frontendOptions,$backendOptions);
```

参数说明：

☑ $frontendName：前端操作模块。

☑ $backendName：后端适配器。



☑　$frontendOptions：前端缓存参数。

☑　$backendOptions：后端缓存参数，主要是设置缓存数据路径。

2．标记记录

标记是一种缓存记录分类的方法。例如，使用 save()方法保存一个缓存，并给该缓存记录设置一个或多个标记，如果是多个标记则以数组形式组织在一起。此后，当不再需要该缓存记录时，就可以根据标记清除所有缓存记录。

```
$cache->save($huge_data, 'myUniqueID', array('tagA', 'tagB', 'tagC'));
$cache->clean(Zend_Cache::CLEANING_MODE_MATCHING_TAG, array('tagA', 'tagB', 'tagC'));
```

3．缓存清理

☑　删除特定 id 的缓存记录，使用 remove()方法。

例如，删除指定 id 的缓存记录，代码如下：

```
$cache->remove('idToRemove');
```

☑　在单独的操作中删除多个缓存记录，使用 clean()方法。

例如，删除所有的缓存记录，代码如下：

```
$cache->clean(Zend_Cache::CLEANING_MODE_ALL);
```

例如，删除过期缓存记录，代码如下：

```
$cache->clean(Zend_Cache::CLEANING_MODE_OLD);
```

例如，删除指定标记 tagA 和 tagC 的缓存记录，代码如下：

```
$cache->clean(Zend_Cache::CLEANING_MODE_MATCHING_TAG, array('tagA', 'tagC'));
```

在本实例中，设置缓存生命周期为 30 秒，将缓存文件存储于根目录下的 tmp 文件夹下，同时输出缓存时间和当前时间。运行结果如图 18.17 所示。



图 18.17　比较缓存时间和当前时间

代码如下：

```
<?php
header("Content-Type: text/html; charset=utf-8");
$frontendOptions = array(
    'lifeTime' => 60,                                                   //设置缓存时间为 30 秒
    'automatic_serialization' => false                                  //关闭自动序列化
);
$backendOptions = array('cache_dir' => './tmp/');                       //设置缓存路径
$arrayIncludePath = array('.', realpath(dirname(__FILE__) . '/../../../library'));  //指定工程包含目录
set_include_path(implode(PATH_SEPARATOR, $arrayIncludePath));           //将指定路径包含到工程中
include 'Zend/Cache.php';                                               //包含文件
$cache = Zend_Cache::factory('Output',
                             'File',
                             $frontendOptions,
                             $backendOptions);
//传递一个唯一标识符给 start()方法，此方法为从 start 之后到 end 结束之前的所有数据都会被缓存
```

```
if(!$cache->start('mypage')) {
    echo '缓存中的时间为：'. date("Y-m-n H:i:s"). '<br>';
    $cache->end();
}
echo '本地时间为：'. date("Y-m-n H:i:s");
//$cache->clean(Zend_Cache::CLEANING_MODE_ALL);          //清除所有缓存
//$cache->clean(Zend_Cache::CLEANING_MODE_OLD);          //清除过期缓存
```

照猫画虎：应用 Zend_Cache 的前端操作缓存程序所有页面，其中 login 页面不缓存，在每个页面中都输出当前时间。(20 分)(实例位置：光盘\mr\18\zmhh\02_zmhh)

18.8.3　基本功训练 3——应用 Zend_Config 配置站点初始参数

视频讲解：光盘\mr\18\lx\应用 Zend_Config 配置站点初始参数.exe

实例位置：光盘\mr\18\zmhh\03

为了使站点正常运行，必须告诉它使用的相关信息，如数据库名称、用户名和密码等。因为不希望将这些信息直接编码（hard-code）到程序中，所以使用配置文件来保存这些信息。

Zend Framework 提供一个 Zend_Config 类，以面向对象的方式来访问配置文件。配置文件可以是 INI 或者 XML 文件。本实例使用 INI 格式将配置信息保存在 application/config 目录下的 application.ini 文件中，运行结果如图 18.18 所示。

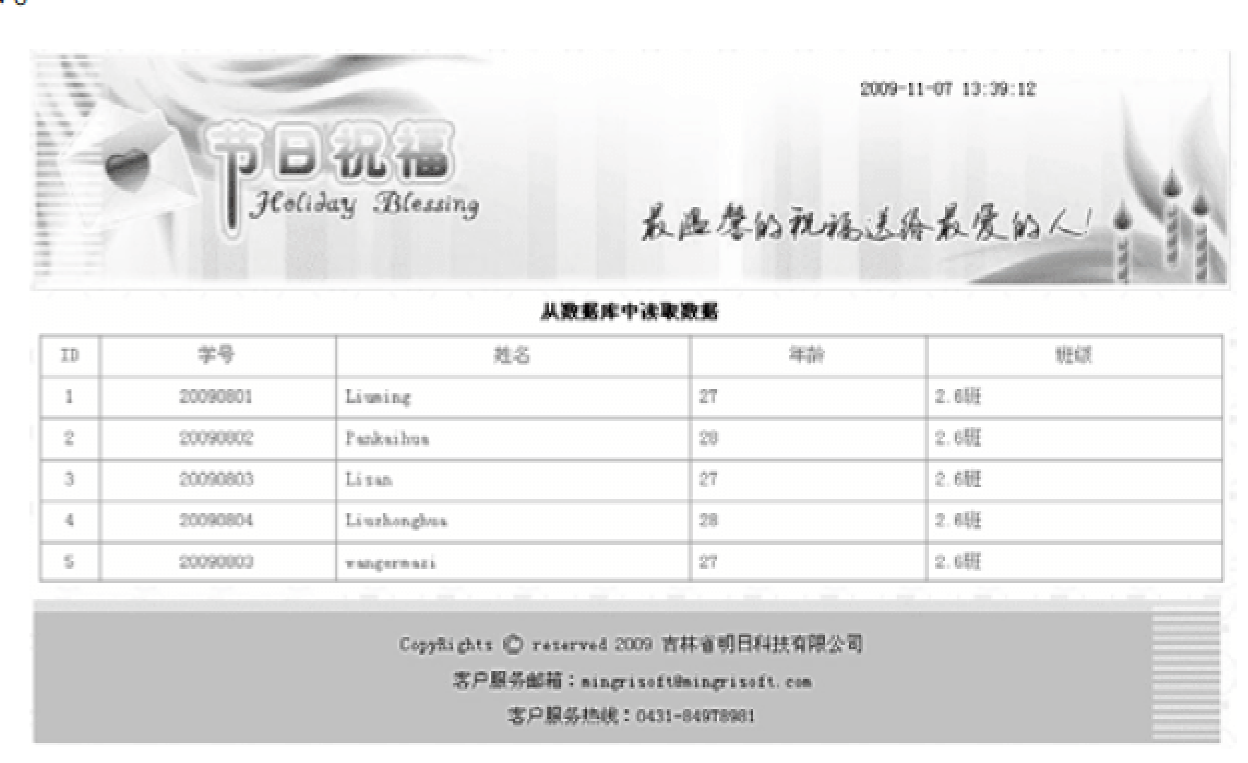


图 18.18　数据库中读取数据

在本实例中将连接数据库所需数据定义到配置文件中，通过配置文件连接数据库并将数据库中的数据输出。创建配置文件的步骤如下：

（1）创建配置文件 application/config/application.ini，写入配置信息。

（2）通过配置文件的头部字母解析配置文件。

（3）运行站点，查看配置文件是否正确，是否正确解析配置文件。

Zend_Config 可以基于数据创建面向对象的 wrapper，格式如下：

```
New Zend_Config(ConfigArray)
```

参数 ConfigArray 为配置数据的数组。

当然 Zend_Config 配置站点也可以通过运行 Zend_Config_Ini($filename, $section, $options = false)来实现读取配置文件。Zend_Config_Ini 的参数说明如表 18.6 所示。



表 18.6　Zend_Config_Ini 参数说明

值	说　明
$filename	要加载的 INI 文件
$section	在 INI 文件中[section]（节）将被加载。把这个参数设置为 null，所有的节将被加载。另外，一个节名称的数组被提供给加载多个节
$options = false	选项数组。下面的键被支持。 ☑ allowModifications：设置为 true 允许随后加载文件更改。默认为 false ☑ nestSeparator：设置嵌套字符。默认为"."

如果使用的是 Zend Framework 1.8.0 以上版本，可以通过 Zend_Application($section，$filename)来实现读取配置文件，参数如表 18.6 所示。

Zend_Config 配置站点初始参数的步骤如下：

（1）搭建 Zend Framework MVC 环境。

（2）创建 application/config/application.ini 文件，代码如下：

```
[project]
bootstrap.path = "../application/Bootstrap.php"
bootstrap.class = "Bootstrap"
phpSettings.display_errors = 1
phpSettings.date.timezone = "prc"
resources.frontController.controllerDirectory = "../application/controllers"
resources.layout.layoutPath = "../application/layouts"
resources.db.adapter = "PDO_MYSQL"
resources.db.params.host = "localhost"
resources.db.params.username = "root"
resources.db.params.password = "111"
resources.db.params.dbname = "db_database18"
resources.db.params.driver_options.1002 = "set names utf8"
```

配置文件内容分析如下：

bootstrap.path 为调用启动文件的路径，class 为调用启动文件的类。

display_errors 为返回错误信息，date.timezone 为设定的时间格式。

frontController 为控制器（C 层）路径。

layoutPath 为布局文件路径。

adapter 为使用 MySQL 数据库。

host 为数据库服务器，username 为数据库用户名，password 为数据库密码。

dbname 为数据库库名。

driver_options 为数据库存储数据时存入 utf8 编码。

（3）在 html 文件创建的 index.php 下加入如下代码：

```
$application = new Zend_Application('project','../application/configs/application.ini');
```

（4）在 models/DbTable 文件下创建 Students.php，定义读取数据库中数据和缓存文件名称，代码如下：

```
class Model_DbTable_Students extends Zend_Db_Table
{
      protected $_name = 'tb_students';
      protected $_primary = 'id';
```

```
        private $_adapter;

        public function init()
        {
                $this->_adapter = $this->getAdapter();
        }
}
```

其中，$_name 为数据库中表的名称，$_primary 为数据库中主键名称，$_adapter 为所有的 Zend_Db_Table 默认。

（5）在 controllers 文件下创建 IndexController.php，定义 V 层中 index.phtml 的输出内容，代码如下：

```
class IndexController extends Zend_Controller_Action
{
        public function init()
        {
                $this->_helper->layout->setLayout("default");
                $this->view->assign('title', 'Zend_Config 配置站点初始参数');
                $this->view->assign('datetime', date('Y-m-d H:i:s'));
        }

        public function indexAction()
        {
                $studnets = new Model_DbTable_Students();

                $this->view->assign('content', '从数据库中读取数据');
                $this->view->studentsModel = $studnets->fetchAll();
        }
}
```

（6）在 views/scripts/index 文件下创建 index.phtml，输出$content 和$studentrsModel，代码如下：

```
<?php echo $this->escape($this->content); ?>

<?php
        foreach ( $this->studentsModel as $tb_student ) :
?>
        <tr>
                        <td align="center" bgcolor="#FFFFFF" class="STYLE1"><?php
                echo $this->escape ( $tb_student->id );
                ?></td>
                        <td align="center" bgcolor="#FFFFFF" class="STYLE1"><?php
                echo $this->escape ( $tb_student->num );
                ?></td>
                        <td align="left" bgcolor="#FFFFFF" class="STYLE1"
                                style="padding-left: 5px; padding-top: 5px; padding-bottom: 5px; padding-right: 5px;">
<?php
                echo $this->escape ( $tb_student->name );
                ?></td>
                        <td align="left" bgcolor="#FFFFFF" class="STYLE1"
                                style="padding-left: 5px; padding-top: 5px; padding-bottom: 5px; padding-right: 5px;">
<?php
                echo $this->escape ( $tb_student->age );
                ?></td>
```

```
                <td align="left" bgcolor="#FFFFFF" class="STYLE1"
                    style="padding-left: 5px; padding-top: 5px; padding-bottom: 5px; padding-right: 5px;"> <?php
            echo $this->escape ( $tb_student->class );
            ?>班</td>
        </tr>
<?php
        endforeach;
?>
```

照猫画虎：根据 Zend_Config_Ini 配置站点初始参数。（20 分）（实例位置：光盘\mr\18\zmhh\03_zmhh）

18.8.4 基本功训练 4——应用 Zend_Form 制作用户注册表单

视频讲解：光盘\mr\18\lx\应用 Zend_Form 制作用户注册表单.exe

实例位置：光盘\mr\18\zmhh\04

Zend_Form 是 Zend Framework 自带组件，为 Web 程序中简化表单创建和处理，它可以将输入的元素过滤和校验，还可以对表单元素进行排序、解析。运行本实例可以看到完整的 Zend_Form 用户注册表单，如图 18.19 所示。



图 18.19 Zend_Form 用户注册表单

本实例应用 Zend_Form 的校验器、过滤器和创建表单属性，实现用户注册功能。下面逐一介绍这些常用属性。

☑ 校验器：根据要求检查它的输入，并返回一个布尔型结果。如果输入信息不符合要求，校验器另外提供信息来说明输入不符合要求。

```
SetRequired($flag)
```

设置和读取 required 标志的状态。当设置为 true，这个标志要求元素在由 Zend_Form 处理的数据中。

```
addValidator($nameOrValidator, $breakChainOnFailure = false, array $options = null)
```

当设置为 NotEmpty 时，就表示该输入不能为空。

☑ 过滤器：在校验之前对输入执行规范化。

```
addFilter($nameOfFilter, array $options = null)
```

在本实例中使用的是短过滤器名，短过滤器名一般就是去掉 Zend_Filter_前缀，首字母不需要大写。

☑ 创建表单属性：对于表单元素，属性的指定由一组访问器来完成。

具体实现步骤如下：

（1）搭建 Zend Framework MVC 环境。

（2）Zend_Config 配置站点初始参数。

（3）在 application/models/DbTable 文件下建立 StudentForm.php 文件，建立 Zend_Form，确定数据库表名，设置表单元素。代码如下：

```
class Model_DbTable_StudentForm extends Zend_Form
{
	public function __construct($option = null)
	{
		parent::__construct($option);
		$this->setName('tb_students');
		$id = new Zend_Form_Element_Hidden('id');
		$num = new Zend_Form_Element_Text('num');
		$num->setLabel('学号')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$name = new Zend_Form_Element_Text('name');
		$name->setLabel('姓名')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$age = new Zend_Form_Element_Text('age');
		$age->setLabel('年龄')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$class = new Zend_Form_Element_Text('class');
		$class->setLabel('班级')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$submit = new Zend_Form_Element_Submit('submit');
		$submit->setAttrib('id','submitbutton');
		$this->addElements(array($id,$num,$name,$age,$class,$submit));
	}
}
```

（4）在 application/controllers 文件下建立 IndexController.php，再建立 Form，并以 POST 方式得到表单中的元素，存入数据库中，返回本页面。代码如下：

```
public function indexAction()
	{
		$form = new Model_DbTable_StudentForm();
		$form->submit->setLabel('注 册');
		$this->view->form = $form;
		if ($this->_request->isPost())
		{
			$formData = $this->_request->getPost();
			if ($form->isValid($formData))
```

```
            {
                $students = new Model_DbTable_Students();
                $row = $students->createRow();
                $row->num = $form->getValue('num');
                $row->name = $form->getValue('name');
                $row->age = $form->getValue('age');
                $row->class = $form->getValue('class');
                $row->save();
                $this->_redirect('/');
            }else
            {
                $form->populate($formData);
            }
        }
```

（5）在 application/views/scripts/index 下建立 index.phtml 文件，文件中只需要调用控制层中的显示 form 即可实现。代码如下：

```
<?php echo $this->form; ?>
```

注意：Model 层中的文件如果类的名称前加入 DbTable，则相应地在 models 文件下建立 DbTable 文件，将类文件建在下面；反之如果没有，则不必建立。

照猫画虎：根据上述介绍的 Zend_Form 组件，设计用户登录页面。（20 分）（实例位置：光盘\mr\18\zmhh\04_zmhh）

18.8.5　基本功训练 5——应用 Zend_Mail 发送邮件

视频讲解：光盘\mr\18\lx\应用 Zend_Mail 发送邮件.exe

实例位置：光盘\mr\18\zmhh\05

Zend Framework 通过 Zend_Mail 组件发送邮件，邮件在 Zend_Mail 中通过默认的 Zend_Mail_Transport_Sendmail 和 Zend_Mail_Transport_Smtp 来发送。在本实例中通过 Zend_Mail_Transport_Smtp 来发送邮件，运行结果如图 18.20 所示。



图 18.20　Zend_Mail 发送邮件

发送邮件应用 Zend_Mail_Transport_Smtp($IP)，$IP 地址为服务器 IP 地址。Zend_Mail 发送邮件有 4 个基本属性，如表 18.7 所示。

表 18.7 Zend_Mail 发送邮件的 4 个基本属性

值	说 明
addTo($option,$to)	收件人地址，$option 为发件人地址，$to 为邮件中发件人地址显示
serFrom($option,$from)	发件人地址，$option 为收件人地址，$from 为邮件中收件人地址显示
setSubject($subject)	邮件标题
setbodyText($body)	邮件内容

Zend_Mail 发送邮件的具体步骤如下：

（1）搭建 Zend Framework MVC 环境。

（2）Zend_Config 配置站点初始参数。

（3）在 application/models/DbTable 文件下创建 MailForm.php 文件，创建表单收件人、发件人、标题和内容。代码如下：

```
class Model_DbTable_MailForm extends Zend_Form
{
	public function __construct($option = null)
	{
		parent::__construct($option);
		$this->setName('form1');
		// form 控制器需要知道有哪些内容
		$to = new Zend_Form_Element_Text('to');
		$to->setLabel('收件人')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$from = new Zend_Form_Element_Text('from');
		$from->setLabel('发件人')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$subject = new Zend_Form_Element_Text('subject');
		$subject->setLabel('标题')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$content = new Zend_Form_Element_Textarea('content');
		$content->setLabel('内容')
			->setRequired(true)
			->addFilter('StripTags')
			->addFilter('StringTrim')
			->addValidator('NotEmpty');
		$submit = new Zend_Form_Element_Submit('submit');
		$submit->setAttrib('id','submitbutton');
		$this->addElements(array($to,$from,$subject,$content,$submit));
	}
}
```

（4）在 application/controllers 文件下创建 IndexController.php，确定使用类及服务器 IP 地址，实例化 Zend_Mail，填入表单元素，发送邮件。代码如下：

```
public function indexAction()
{
        $form = new Model_DbTable_MailForm();
        $form->submit->setLabel('发  送');
        $this->view->form = $form;
        if ($this->_request->isPost())
        {
                $formData = $this->_request->getPost();
                if ($form->isValid($formData))
                {
                        $tr = new Zend_Mail_Transport_Smtp('192.168.1.247');
                        $mail = new Zend_Mail();

                        $mail->addTo('cym3100@163.com',$form->getValue('to'));
                        $mail->setFrom('cym3100@163.com',$form->getValue('from'));
                        $mail->setSubject($form->getValue('subject'));
                        $mail->setBodyText($form->getValue('content'));
                        $mail->send($tr);

                        $this->_redirect('/');
                }else
                {
                        $form->populate($formData);
                }
        }
}
```

（5）在 application/views/scripts/index 文件下创建 index.phtml，在该页面中调用 Form 表单。代码如下：

```
<?php echo $this->form; ?>
```

说明：通过 Zend_Mail 发送邮件需要实例化 Zend_Mail_Transport_Smtp($ip)，这个 IP 地址必须为服务器 IP。AddTo()和 AddFrom()中的第一个参数必须是服务器中的一个邮箱名称。

照猫画虎：在上述实例基础上实现邮件的群发功能。（20 分）（实例位置：光盘\mr\18\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

18.9　情景应用——拓展与实践

18.9.1　情景应用 1——应用 Zend Framework 进行用户身份验证

视频讲解：光盘\mr\18\lx\应用 Zend Framework 进行用户身份验证.exe

实例位置：光盘\mr\18\qjyy\01

使用 Zend Framework 框架实现用户身份的验证主要应用 Zend_Auth 组件。明日 IT 新闻网的用户登录模

块应用的就是 Zend_Auth 组件对用户身份进行验证。运行结果如图 18.21 所示。



图 18.21　明日 IT 新闻网的登录页面

Zend_Auth 组件实现对明日 IT 新闻网的用户身份验证，其具体步骤如下：

（1）建立 Zend Framework 的 MVC 框架结构。

（2）建立用户登录表单。该表单中的元素包括用户昵称文本框、用户密码文本框和验证码文本框，实现该过程的代码主要由 HTML 语句的表单相关标签组成，具体实现代码详见本书附带光盘，其运行结果如图 18.21 所示。

（3）在用户表模型中建立 isValid()方法，用于验证登录信息是否正确，正确则返回 TRUE，反之返回 FALSE。application\models\DbTable\News.php 文件的关键代码如下：

```
public function isValid ($netname, $password)
{
    //构建验证适配器
    $authAdapter = new Zend_Auth_Adapter_DbTable($this->getAdapter());
$authAdapter->setTableName($this->_name)->setIdentityColumn('netname')->setCredentialColumn('password')
->setIdentity($netname)->setCredential($password);
    //获取 Zend_Auth 对象并验证
    $auth = Zend_Registry::get('auth');
    return $auth->authenticate($authAdapter)->isValid();
}
```

（4）当用户在用户登录表单中录入用户信息并单击“登录”按钮时，登录信息将被提交到 UserController 控制器的 loginAction 动作中验证用户的登录信息是否正确。application\modules\default\controllers\UserController.php 文件的代码如下：

```
if ($this->_request->isPost()) {
    //接收用户名密码
    $netname = trim($this->_request->getParam('netname'));
    $password = trim($this->_request->getParam('password'));
    //对用户身份进行验证
    if ($this->_user->isValid($netname, md5($password))) {
        $url = $this->_request->getParam('url');
        if ($url == null) {
            //定位到登录成功页面
            $this->_redirect('/user/success/flag/login');
        } else {
            $url = str_replace('@', '?', str_replace('$', '&', $url));
            $this->_redirect($url);
        }
```

```
            exit();
        } else {
            $errMsg = '登录昵称或密码输入有误，请重新登录！';
            $this->view->errMsg = $errMsg;
            $this->view->netname = $netname;
        }
}
```

DIY：通过 Zend Framework 实现用户注册的功能。（20 分）（实例位置：光盘\mr\18\qjyy\01_diy）

18.9.2 情景应用 2——应用 Zend Framework 实现查询结果关键字描红

视频讲解：光盘\mr\18\lx\应用 Zend Framework 实现查询结果关键字描红.exe

实例位置：光盘\mr\18\qjyy\02

Zend Framework 框架是一个扩展性非常强大的 MVC 框架，开发人员可以定义自己的视图助手、校验器等。在本实例中通过自定义视图助手实现在明日 IT 新闻网中查询结果关键字的描红输出，其运行结果如图 18.22 所示。



图 18.22 明日 IT 新闻网的新闻查询模块

具体步骤如下：

（1）建立 Zend Framework 的 MVC 框架结构。

（2）建立查询表单，如图 18.22 所示，该过程的详细代码详见本书附带光盘。

（3）在新闻表模型中建立方法，实现对新闻信息的匹配查询。application\models\DbTable\News.php 文件的代码如下：

```
public function findByLike ($keywords)
{
    //将关键字保存到数组中
    $arrayKeyWords = explode(' ', $keywords);
    //查询结果排序方式
    $order = 'addtime desc';
    //构建 select 对象
    $select = $this->getAdapter()->select();
    //指定查询表名
    $select->from($this->_name);
    //指定查询条件
    $orWhere = '';
    $j = 0;
    //按主题查询
    foreach ($arrayKeyWords as $key) {
        if (trim($key) != '') {
            if ($j == 0) {
                $orWhere .= $this->getAdapter()->quoteInto('title like ?', '%' . $key . '%');
            } else {
```

```
                    $orWhere .= $this->getAdapter()->quoteInto(' or title like ?', '%' . $key . '%');
                }
                $j ++;
            }
        }
        //按内容查询
        foreach ($arrayKeyWords as $key) {
            if (trim($key) != '') {
                if ($j == 0) {
                    $orWhere .= $this->getAdapter()->quoteInto('content like ?', '%' . $key . '%');
                } else {
                    $orWhere .= $this->getAdapter()->quoteInto(' or content like ?', '%' . $key . '%');
                }
                $j ++;
            }
        }
        $select->where($orWhere);
        //排序方式
        $select->order($order);
        $result = $this->getAdapter()->fetchAll($select);
        return $result;
}
```

（4）在 CommonController 控制器的 searchAction 动作中对用户提交关键字进行查询。application\modules\default\controllers\CommonController.php 文件的代码如下：

```
public function searchAction ()
{
        //获取页面提交参数
        $keyWord = urldecode($this->_request->getParam('keyWord'));
        $page = $this->_request->getParam('page');
        $this->view->keyWord = urlencode($keyWord);
        //关键字数组
        $keywordsStrArray = explode(' ', $keyWord);
        $this->view->keywordsStrArray = $keywordsStrArray;
        //开始查询
        $array = array();
        //新闻
        $model = new Model_DbTable_News();
        $infos = $model->findByLike($keyWord);
        $tmpArray = array();
        foreach ($infos as $info) {
            $tmpArray['flag'] = 0;
            $tmpArray['id'] = $info['id'];
            $tmpArray['title'] = $info['title'];
            $tmpArray['uncontent'] = $info['uncontent'];
            $tmpArray['addtime'] = $info['addtime'];
            array_push($array, $tmpArray);
        }
        //分页参数
        if ($page == null) {
            $page = 1;
```

```
    }
    $pageRange = 10;
    $itemCountPerPage = 10;
    //实例化并构建 Zend_Paginator 对象
    $paginatorAdapter = new Zend_Paginator_Adapter_Array($array);
    $paginator = new Zend_Paginator($paginatorAdapter);
$paginator->setPageRange($pageRange)->setItemCountPerPage($itemCountPerPage)->setCurrentPageNum
ber($page);
    $this->view->paginator = $paginator;
    //设置布局
    $this->_helper->layout->disableLayout();
    //页面信息
    $this->view->title = '内容查询-' . $this->_config['pageInfo']['default']['title'];
}
```

（5）在 Scripts 目录的 helpers 子目录中，编写用于实现关键字描红的视图助手 Zend_View_Helper_SetRed。application\modules\default\views\helpers\SetRed.php 文件的代码如下：

```
class Zend_View_Helper_SetRed
{
    public function setRed ($arrayKeywords, $str)
    {
        $beginFont = '︼';                                    //关键字开始标识
        $endFont = '︻';                                      //关键字结束标识
        foreach ($arrayKeywords as $keyword) {                //遍历所有关键字
            $array = array();
            preg_match_all('/' . $keyword . '/i', $str, $array);   //查询与关键字匹配的内容，并保存到数组中
            $arrayKey = array_unique($array[0]);              //去掉重复数组元素
            foreach ($arrayKey as $key) {                     //遍历所有匹配关键字
//在查询结果中的匹配关键字前后分别加上关键字开始标识和结束标识
                $str = str_replace($key, $beginFont . $key . $endFont, $str);
            }
        }
//将开始标识替换成<font>标签开始部分
        $str = str_replace($beginFont, '<font color="#FF0000">', $str);
        $str = str_replace($endFont, '</font>', $str);        //将结束标识替换成<font>标签结束部分
        return $str;
    }
}
```

（6）完成以上步骤就可以在视图中通过 foreach 循环语句显示查询结果，在视图文件中通过$this 关键字调用视图助手。application\modules\default\views\scripts\common\search.phtml 文件的代码如下：

```
<?php if(count($this->paginator)>0):?>
<div class="nav">
    <?php  echo  $this->paginationControl($this->paginator,  'Sliding',  'list_search_pagination_control.phtml',
array('keyWord'=>$this->keyWord))?>
</div>
<div class="cell_h"></div>
<?php foreach ($this->paginator as $search):?>
<div class="search">
    <div class="t">
        <a href="<?php echo $this->baseUrl($this->searchLink($search['id'], $search['flag'], 0, $search['user_
id']))?>"  target="_blank"  class="a25"><?php  echo  $this->setRed($this->keywordsStrArray,  $this->escape
```

```
($search['title']))?></a>
    </div>
    <div class="c">
        <?php echo $this->setRed($this->keywordsStrArray, $this->substr($search['uncontent'], 380))?>  <a
href="<?php echo $this->baseUrl($this->searchLink($search['id'], $search['flag'], 0))?>" target="_blank"  class
="a24">&lt;详细&gt;</a>
    </div>
    <div class="i">
        <font color="#339900"><?php echo $this->searchLink($search['id'], $search['flag'], 1, $search['user_
id'])?></font>  <?php echo substr($search['addtime'], 0, 16)?>
    </div>
</div>
<br/>
<?php endforeach;?>
```

DIY：将上述查询结果中的数据进行分页显示。（20 分）（实例位置：光盘\mr\18\qjyy\02_diy）

18.9.3　情景应用 3——应用 Zend Framework 设计新闻发布系统首页

视频讲解：光盘\mr\18\lx\应用 Zend Framework 设计新闻发布系统首页.exe

实例位置：光盘\mr\18\qjyy\03

新闻发布系统的首页以突出多样化的新闻栏目为主，重点突出今日的重大新闻。新闻系统的首页设计效果如图 18.23 所示。



图 18.23　新闻系统的首页设计效果

新闻发布系统首页默认控制器是 NewsController，在 NewsController 控制器中定义 indexAction、listAction 和 infoAction 3 个动作方法，调用新闻、评论和用户表模型中的方法完成数据的读取操作，并将读取到的数



据传递到视图中，最终，在视图中输出首页的数据。

（1）创建 NewsController 控制器，application\modules\default\controllers\NewsController.php 文件的关键代码如下：

```
<?php
/*  新闻控制器  */
class NewsController extends Zend_Controller_Action{
      private $_config;                                                          //项目配置信息
      private $_news;                                                            //新闻表模型
      private $_user;                                                            //用户表模型
      private $_comment;                                                         //评论表模型
      private $_auth;                                                            //Zend_Auth 对象
private $_loginUser;                                                             //当前登录用户
      /**初始控制器*/
      public function init (){
            $this->initView();                                                   //初始化视图
            $this->_config = Zend_Registry::get('config');                       //初始化工程配置信息对象
            $this->_news = new Model_DbTable_News();                             //实例化新闻表模型
            $this->_user = new Model_DbTable_User();                             //实例化用户表模型
          //省略了部分代码
      }
      /*  设置页面信息*/
      private function _setPageInfo ($title = '', $keywords = '', $description = '', $position = '',   $layout = 'default'){
            $this->view->title = $title;                                         //主题
            $this->view->keywords = $keywords;                                   //关键字
            $this->view->description = $description;                             //描述
            $initPosition = '<a href="' . $this->view->baseUrl('/') . '" class="a2">' . $this->_config['pageInfo']['default']
['title'] . '</a>    >>    <a href="' . $this->view->baseUrl('/news') . '" class="a2">' . $this->_config['pageInfo']['news']['title'] .
'</a>';
            $this->view->position = $initPosition . $position;                   //导航
            //布局
            if ($layout == null) {
                  $this->_helper->layout->disableLayout();
            } else {
                  $this->_helper->layout->setLayout($layout);
            }
      }
      /**根据标识获取新闻类别名称*/
      private function _getNewsTypeByFlag ($flag){
            $typeName = '';
            switch ($flag) {
                  case 'ttj':
                        $typeName = '今日推荐新闻';
                        break;
                  case 'all':
                        $typeName = '全部新闻';
                        break;
                  //省略了部分代码
                  case 10:
                        $typeName = '综合资讯';
                        break;
```

```
        }
        return $typeName;
    }
    /**新闻首页 Action*/
    public function indexAction (){
        //今日推荐新闻
        $todayCommendNewses = $this->_news->findByNum('istodaycommend', 0, 0, 10, false, true, true);
        $this->view->todayCommendNewses = $todayCommendNewses;
        //省略了部分代码
        //设置页面信息
        $title = $this->_config['pageInfo']['default']['title'];
        $keywords = $this->_config['pageInfo']['news']['keywords'];
        $description = $this->_config['pageInfo']['news']['description'];
        $this->_setPageInfo($title, $keywords, $description, '');
    }
    /**新闻列表页 Action*/
    public function listAction (){
        //接收参数
        $flag = $this->_request->getParam('flag');
        $order = $this->_request->getParam('order');
        $page = $this->_request->getParam('page');
        $itemCountPerPage = 10;   //分页参数
        $pageRange = 5;
        $this->view->flag = $flag;   //为视图变量赋值
        $this->view->order = $order;
        //新闻分页
        $paginator = $this->_news->findByPage($flag, $order, $page, $itemCountPerPage, $pageRange);
        $this->view->paginator = $paginator;
        //省略了部分代码
        //设置页面信息
        $title  =  $this->_getNewsTypeByFlag($flag)  .  '-'  .  $this->_config['pageInfo']['news']['title']  .  '-'  .
$this->_config['pageInfo']['default']['title'];
        $keywords = $this->_getNewsTypeByFlag($flag);
        $description = '';
        $position = '  >>  <a href="' . $this->view->baseUrl('/news/list-' . $flag . '-0-1.html') . '" class="a2">' .
$this->_getNewsTypeByFlag($flag) . '</a>';
        $this->_setPageInfo($title, $keywords, $description, $position);
    }
    /**新闻详细信息页 Action*/
    public function infoAction (){
       //省略了部分代码
    }
}
```

（2）在新闻表（tb_news）模型 Model_DbTable_News 中建立 findByNum()方法，查询新闻表中的数据。application\models\DbTable\News.php 文件的关键代码如下：

```
public function findByNum ($flag, $orderIndex, $offset, $count, $isImg = false, $isCache = false, $affect = false){
    $select = $this->getAdapter()->select();                    //生成 select 对象
    $select->from($this->_name);
    //isTodayCommend-今日推荐  all-全部  0-站内新闻   1-IT 人物  2-高端访谈  3-产品快讯  4-企业动态
5-互联网  6-游戏资讯  7-广告传媒  8-财经报道  9-图片新闻  10-综合资讯
```

```
    if ($flag !== 'all') {
        if ($flag === 'istodaycommend') {
            $where = 'istodaycommend = true';
        } else {
            $where = $this->getAdapter()->quoteInto('flag = ?', $flag);
        }
        $select->where($where);
    }
    //只显示带图片的记录
    if ($isImg) {
        $select->where("imagename <> ''");
    }
    //排序方式
    if ($orderIndex == 0) {
        $order = 'addtime desc';
    } elseif ($orderIndex == 1) {
        $order = 'addtime';
    } elseif ($orderIndex == 2) {
        $order = 'browse desc';
    } elseif ($orderIndex == 3) {
        $order = 'browse ';
    }
    $select->order($order);
    $select->limit($count, $offset);                    //指定记录条数
    if ($isCache) {                                     //查询结果
        $cacheId = strtolower($this->_name . '_' . __FUNCTION__ . '_' . $flag . '_' . $orderIndex . '_' . $offset .
'_' . $count . '_' . $isImg);                           //缓存 ID
        $arrayCachelag = array();                       //缓存标识
        if ($affect) {
            $arrayCachelag = array(strtolower($this->_name));
        }
        //带缓存的查询
        if (! $result = $this->_cache->load($cacheId)) {
            $result = $this->getAdapter()->fetchAll($select);
            $this->_cache->save($result, $cacheId, $arrayCachelag);
        }
    } else {
        //不带缓存的查询
        $result = $this->getAdapter()->fetchAll($select);
    }
    return $result;
}
```

（3）创建与 NewsController 控制器对应的视图，输出首页中的数据。其中今日推荐模块 application\modules\default\views\scripts\news\index.phtml 文件的关键代码如下：

```
<div class="center">
        <ul>
            <li class="li1">
                [今日推荐]
            </li>
        <?php foreach ($this->todayCommendNewses as $key => $todayCommendNews):?>
```

```
                <?php if($key < 2):?>
                <li class="li2"><a href="<?php echo $this->baseUrl('/news/info-'.$todayCommendNews ['id'].'-1.html')?>" title="<?php echo $this->escape($todayCommendNews['title'])?>" class="<?php if($key == 0):?>a5<?php else:?>a6<?php endif;?>"><?php echo $this->substr($this->escape ($todayCommendNews ['title']), 50)?></a><br/>[<?php echo $this->substr($todayCommendNews['uncontent'], 60)?>]</li>
                <?php else:?>
                <li class="li3">¡¤<a href="<?php echo $this->baseUrl('/news/info-'.$todayCommendNews ['id'].'-1.html')?>" title="<?php echo $this->escape($todayCommendNews['title'])?>"><?php echo $this->substr ($this->escape ($todayCommendNews['title']), 60)?></a></li>
                <?php endif;?>
            <?php endforeach;?>
            </ul>
</div>
```

DIY：根据新闻发布系统首页中展示的新闻超链接，完成新闻详细内容查看功能。（20 分）（实例位置：光盘\mr\18\qjyy\03_diy）

18.9.4　情景应用 4——应用 Zend_Paginator 实现数据分页显示

视频讲解：光盘\mr\18\lx\应用 Zend_Paginator 实现数据分页显示.exe

实例位置：光盘\mr\18\qjyy\04

本实例应用 Zend_Paginator 对数据库中的数据进行分页显示，运行结果如图 18.24 所示。



图 18.24　Zend_Paginator 分页显示

具体步骤如下：

（1）搭建 Zend Framework MVC 环境。

（2）通过 Zend_Config 配置站点初始参数。

（3）在 application\controllers\IndexController.php 中加入如下代码：

```
        $pageNumber = 5;
        $paginator = Zend_Paginator::factory($this->view->studentsModel);
    $paginator->setItemCountPerPage($pageNumber);
    $paginator->setCurrentPageNumber($this->_getParam('page'));
    $paginator->setPageRange(5);
    Zend_Paginator::setDefaultScrollingStyle('Sliding');
    $paginator->setView($this->view);
```

```
        $this->view->studentsModel = $paginator;
        $this->view->paginator = $paginator;
        $this->render();
```

其中，Zend_Paginator::factory();为分页器的工厂方法，参数为数组中读取出的数据（数组）；Zend_Paginator::setDefaultScrollingStyle('Sliding')为调用页码样式；$this->view->paginator 将分页传给 V 层。

（4）在 application\views\scripts\index 文件下建立 pagelist.phtml 文件，自己建立所需要的分页页码，代码如下：

```
<?php if ($this->pageCount): ?>
<div class="paginationControl">
<!-- Previous page link -->
<?php if (isset($this->previous)): ?>
   <a href="<?php echo $this->url(array('controller'=>'index', 'action'=>'index', 'page'=>$this->previous)); ?>">< 
上一页</a> |
<?php else: ?>
   <span class="disabled">< 上一页</span> |
<?php endif; ?>
<!-- Numbered page links -->
<?php foreach ($this->pagesInRange as $page): ?>
  <?php if ($page != $this->current): ?>
     <a href="<?php echo $this->url(array('controller'=>'index', 'action'=>'index', 'page'=>$page)); ?>"><?= 
$page; ?></a> |
  <?php else: ?>
    <?= $page; ?> |
  <?php endif; ?>
<?php endforeach; ?>
<!-- Next page link -->
<?php if (isset($this->next)): ?>
   <a href="<?php echo $this->url(array('controller'=>'index', 'action'=>'index', 'page'=>$this->next)); ?>">下一
页 ></a>
<?php else: ?>
   <span class="disabled">下一页  ></span>
<?php endif; ?>
</div>
<?php endif; ?>
```

DIY：在上述内容中，通过修改样式做成下拉菜单分页。(20 分)(实例位置：光盘\mr\18\qjyy\04_diy)

18.9.5　情景应用 5——应用 Zend_Mail 接收邮件

视频讲解：光盘\mr\18\lx\应用 Zend_Mail 接收邮件.exe

实例位置：光盘\mr\18\qjyy\05

通过 POP3 指令对邮箱内的邮件进行查看、删除。在 Zend Framework 中通过 Zend_Mail_Storage_Pop3($array)来对邮箱内的邮件进行管理。但是对于 Zend_Mail 并不能直接应用在服务器中，必须借助第三方软件来实现对邮箱的管理。本实例实现邮件的接收和管理功能，运行结果如图 18.25 所示。

实现对邮件的接收管理，需要应用第三方软件 WinWebMail。下面介绍 WinWebMail 的配置方法。

（1）启动 WinWebMail，用户可以在右下角看到一个带红色 X 的蓝色图标，单击鼠标右键，在弹出的快捷菜单中选择“服务”命令，如图 18.26 所示。



图 18.25　通过 Zend_Mail 接收邮件



图 18.26　WinWebMail 选项

（2）进行如图 18.27 所示的“服务”界面之后，选中“修改”复选框，填入 DNS 的 IP 地址，单击“启动 WinWebMail 服务程序”按钮，单击“绿色对号”，服务选项配置完成。

（3）重新进入图 18.26 所示的“选项”界面，选择“域名管理”命令，添加域名，单击“绿色对号”，进入“域名管理”界面，如图 18.28 所示。



图 18.27　WinWebMail 服务



图 18.28　WinWebMail 域名管理

（4）重新进入图 18.26 所示的“选项”界面，选择“系统设置”命令，添加用户名和密码。选择所在域名。单击“绿色对号”，进入“系统设置”界面，如图 18.29 所示。



图 18.29　WinWebMail 系统设置



至此，WinWebMail 配置完毕。

注意：在使用 WinWebMail 接收邮件之前必须将 Windows 下的 POP3 服务器断开，WinWebMail 不能像自动断开 SMTP 一样断开 POP3 连接。

Zend_Mail 接收邮件的具体步骤如下：

（1）搭建 Zend Framework MVC 环境。

（2）通过 Zend_Config 配置站点初始参数。

（3）在 application\controllers 文件下创建 IndexController.php，将实例化的 POP3 类传入 V 层，代码如下：

```
public function indexAction()
{
    $this->view->mail = new Zend_Mail_Storage_Pop3(array('host' => '192.168.1.247',
                                                          'user' => 'lm',
                                                          'password' => '123'));
}
```

（4）在 application\views\scripts\index 文件下创建 index.phtml，以得到的邮件数量做循环，输出收件人、发件人、标题和内容。代码如下：

```
<?php for ($i = 1;$i <= $this->mail->countMessages();$i++): ?>
<?php $message = $this->mail->getMessage($i); ?>
        <tr>
            <td align="center" bgcolor="#FFFFFF" class="STYLE1"><?php echo $this->escape
($message->to); ?></td>
            <td align="left" bgcolor="#FFFFFF" class="STYLE1"
                style="padding-left: 5px; padding-top: 5px; padding-bottom: 5px; padding-right:
5px;"><?php echo $this->escape($message->from); ?></td>
            <td align="left" bgcolor="#FFFFFF" class="STYLE1"
                style="padding-left: 5px; padding-top: 5px; padding-bottom: 5px; padding-right:
5px;"><?php echo $this->escape($message->subject); ?></td>
            <td align="left" bgcolor="#FFFFFF" class="STYLE1"
                style="padding-left: 5px; padding-top: 5px; padding-bottom: 5px; padding-right:
5px;"><?php echo $this->escape($message->date); ?></td>
        </tr>
<?php endfor; ?>
```

其中，$mail->countMessages()表示邮件数量，$mail->getMessage($i)表示读取第$i 封邮件。

至此，Zend_Mail 接收邮件讲解结束。

DIY：根据上述内容，在接收邮件的同时，完成具体邮件内容的输出。（20 分）（实例位置：光盘\mr\18\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

18.10 自 我 测 试

一、选择题（每题 10 分，5 道题）

1．Zend Framework 框架是由（　　）公司主创。

A．Zend　　B．SUN　　C．IBM

2．由于 Zend Framework 完全采用 PHP 5 的面向对象编程方式实现，所以 Zend Framework 对 PHP 版本的最低要求是（　　）。

A．PHP 5.0　　B．PHP 5.1　　C．PHP 5.2　　D．PHP 5.3

3．MVC 是一种经典的程序设计理念，此模式将应用程序分为 3 个部分，即模型层（Model）、视图层（View）和（　　），MVC 是这 3 个部分英文字母的缩写。

A．控制层（Controller）　　B．管理层（Admin）　　C．中心层（Center）

4．邮件在 Zend_Mail 中通过默认的 Zend_Mail_Transport_Sendmail 和（　　）来发送。

A．Zend_Mail_Transport_Smtp　　B．Zend_Mail_Storage_Pop3

C．Zend_Mail_Storage_Pop4

5．在通过 Zend Framework 框架开发项目时，首先要创建一个 URL 重写文件，这个文件的名称是(　　)。

A．.htaccess　　B．.http　　C．.access

二、填空题（每题 10 分，5 道题）

1．在 Zend Framework 框架中，通过（　　　　）组件操作数据库。

2．在 Zend Framework 框架中，通过（　　　　）组件控制文件的上传。

3．在 Zend Framework 框架中，通过（　　　　）组件进行网站布局。

4．在 Zend Framework 框架中，通过（　　　　）组件发送邮件。

5．在 Zend Framework 框架中，通过（　　　　）组件完成 Web 表单的创建和处理。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

18.11 行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目 分数（　　）	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
		分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。



续表

序号	内　　容	行 动 指 南	
1	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）通过 Zend_Form 组件开发用户注册和登录功能。	
		（2）通过 Zend_Auth 组件完成对登录用户的验证。	
		（3）通过 Zend_Db 组件完成对数据库中数据的操作。	
		（4）通过 Zend_Paginator 组件实现数据库中数据的分页显示。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

18.12　成功可以复制——创造互联网搜索时代拉里·佩奇和谢尔盖·布林

成就：他们创造了谷歌公司，创造了互联网的神话；带来了真正的互联网搜索时代。

创始公司：Google 公司

财富巅峰：两人身价曾并列为全球排名 26 美国排名 12 的亿万富翁。

最高学历：计算机研究生（两人）

出生日期：1973（佩奇）、1974（布林）

超级语录：Google 取得的成功源于其创建者布林和佩奇的想象力，同样也源于他们的天赋。

成功历程

谢尔盖·布林出生在前苏联的一个犹太人家庭。谢尔盖·布林在小学一年级就向老师提交计算机打印输出的设计方案。五岁时跟随父母一起移民美国。中学毕业后，布林进入马里兰大学攻读数学专业，随后进入斯坦福大学读研。在这里他遇到了拉里·佩奇。当时两人都是斯坦福大学的硕士，佩奇 24 岁，布林 23 岁，两人在一次校友会上结识，他们都有很强的主见，意见分歧时互不相让，不管讨论什么问题几乎都会

争吵起来。但最终两人却在解决计算机学最大挑战“搜索引擎”的问题时找到了共同点。

1996 年初，佩奇和布林开始合作研究一名为 BackRub 的搜索引擎，到 1998 年上半年逐步完善这项技术后，两人开始为这项技术寻找合作伙伴。他们找到雅虎的创始人之一戴维·菲洛。菲洛认为他们的技术确实很可靠，但建议他们自己建立一个搜索引擎公司发展业务，发展起来后再考虑合作。吃了无数个闭门羹之后佩奇和布林决定自己创业，但他们手中仅有的一点资金都因购买大量的数据盘和存储器作研究而花光了。他们的一位教授，也是 SUN 微系统的创始人之一安迪·别赫托希姆是个很有远见的人，在看完他们的演示后，立马开了张 10 万美元的支票帮助成立 Google 公司。



Google 搜索引擎源于拉里·佩奇和谢尔盖·布林在斯坦福大学读书时所做的一个研究项目。更确切地说，他们最开始是在佩奇简陋的宿舍搞研究。1998 年 9 月 7 日，Google 公司在加利福尼亚州的曼罗帕克成立，具体说是在一个朋友租给佩奇和布林办公的车库，在当时看来已经不错了，有一台洗衣机，还有热水器。他们最终在小车库搞出了大工程。佩奇说，那时他们俩到处借钱，教授、亲戚、朋友，只要是想得到的人都去借了，筹得 100 万美元作为最初投资。

1999 年 2 月他们搬了新的办公室，虽然条件仍然简陋，但比车库好点，一张乒乓球桌就作为正式的会议场所，8 名员工在办公室里都转不过身，一个人要出门所有人都得起身挪开凳子才能腾出地方。

在这样艰苦的环境下，Google 公司一步步发展。当 Google 的竞争对手致力于成为门户网址、投入搜索服务的比重不大之时，布林反其道而行之，尽力完善其搜索引擎。Google 的使用率越来越高，每天的搜索量由 6 年前的 1 万次增至目前的 3 亿次。很多广告商要求在 Google 网页刊登广告。同时，雅虎、宝洁、美国能源部等许多大公司和政府机构也纷纷使用 Google 的搜索技术，Google 按照搜索施放数来收取授权使用费。Google 公司不仅创造出奇特的搜索引擎技术，更在上市时以拍卖的方式进行定价，完成“对华尔街的清洗”，两个新的亿万富翁就这样诞生了。

深度评价

佩奇和布林的成功告诉我们：要成功，就要勇于创新！而要创新需要一定的灵感，这灵感不是天生的，而是来自长期的积累和全身心的投入。没有积累就不会有创新。所以我们要掌握新技术，要善于学习，更要善于创新！

第19堂课

PHP 的安全与防护

（视频讲解：40 分钟）

随着网络的普及，网上购物已经成了时尚男女的主要消费方式之一。对于个人隐私、账号密码等敏感数据的保护越来越重要。原来只有在侦探小说中才听说过的加密、解密也出现在生活中。其实加密技术本没有那么神秘。它就是一种相对比较复杂的算法。作为普通的开发者来说，可以使用一些已有的、比较有名气的加密算法，如 MD5、SHA 等,也可以自己创建加密函数。

学习摘要：

- PHP 加密函数
- PHP 加密扩展库
- PHP 漏洞和防护措施

19.1 PHP 加密函数

PHP 内置了多种加密函数，这里主要介绍其中的 3 种，即 crypt()函数、md5()函数和 sha1()函数。

19.1.1 使用 crypt()函数进行加密

crypt()函数可以完成单向加密功能。crypt()函数的语法格式如下：

```
string crypt(string str[, string salt]);
```

其中，参数 str 是需要加密的字符串，参数 salt 为加密时使用的干扰串。如果省略第二个参数，则会随机生成一个干扰串。crypt()函数支持 4 种算法和长度，如表 19.1 所示。

表 19.1 crypt()函数支持的 4 种算法和 salt 参数的长度

算 法	salt 长度
CRYPT_STD_DES	2-character（默认）
CRYPT_EXT_DES	9-character
CRYPT_MD5	12-character（以1开头）
CRYPT_BLOWFISH	16-character（以2开头）

说明：默认情况下，PHP 使用一个两个字符的 DES 干扰串，如果系统使用的是 MD5，则会使用一个 12 个字符的干扰串。可以通过 CRYPT_SALT_LENGTH 变量来查看当前所使用的干扰串的长度。

例如，声明一个字符串变量$str，赋值为“This is an example!”，然后使用 crypt()函数进行加密并输出。代码如下：

```
<?php
	$str = 'This is an example!';				//声明字符串变量$str
	echo '加密前$str 的值为：'.$str;
	$crypttostr = crypt($str);				//对变量$str 加密
	echo '<p>加密后$str 的值为：'.$crypttostr;		//输出加密后的变量
?>
```

运行结果如图 19.1 所示。



图 19.1 使用 crypt()函数进行加密

单击“刷新”按钮，会发现每次生成的加密后的结果都不相同，那么问题就出来了：要对加密后的数据进行判断，那该怎么办？crypt()函数是单项加密的，密文不可还原成明码；而每次加密后的数据还不相同，怎么进行判断？

这就是 salt 参数要解决的问题。crypt()函数用 salt 参数对明文进行加密。判断时，对输出的信息再次使



用相同的 salt 参数进行加密，对比两次加密后的结果来进行判断。

例 19.01　对输入的用户名进行检测，如果该用户存在，显示“该用户已存在。”；否则，显示“恭喜您：用户名可以使用!”，代码如下。（**实例位置：光盘\mr\19\sl\19.01**）

```
<?php
	/*	连接数据库	*/
	$conn = mysql_connect("localhost","root","111") or die("数据库连接错误".mysql_error());
	mysql_select_db("db_database19",$conn) or die("数据库访问错误".mysql_error());
	mysql_query("set names utf8");
	/*	**************	*/
?>
<!--	提交表单	-->
<form id="form1" name="form1" method="post" action="">
…
<td><input name="username" type="text" id="username" size="15" /></td>
<td><input type="submit" name="Submit" value="检查" id="Submit" /></td>
…
</form>
<!--	-------------	-->
<?php
	if(isset($_POST['username'])){						//trim()函数去掉字符串两边的空格
		$usr = crypt(trim($_POST['username']),"tm");			//对用户名进行加密
		$sql = "select * from tb_user where username = '".$usr."'";	//生成查询语句
		$rst = mysql_query($sql,$conn);					//执行语句，返回结果集
		$count=mysql_num_rows($rst);
		if($count>0){							//如果结果集大于 0
			echo "<font color='red'>用户名已存在。</font>";		//说明用户名存在
		}else{								//否则说明该用户名可用
			echo "<font color='green'>恭喜您：用户名可以使用!</font>";
		}
	}
?>
```

运行结果如图 19.2 所示。



图 19.2　使用 crypt()函数进行数据验证

19.1.2　使用 md5()函数进行加密

使用 crypt()函数保护重要数据并不是最安全的，只能用在对安全性要求较低的系统中。如果想要提高安全性，则需要使用 md5()函数。

md5()函数使用 MD5 算法。MD5 的全称是 Message-Digest Algorithm 5（信息-摘要算法）。它的作用是把不同长度的数据信息经过一系列的算法计算成一个 128 位的数值，就是把一个任意长度的字节串变换成

一定长的整数。请注意这里是“字节串”而不是“字符串”这个词，是因为这种变换只与字节的值有关，与字符集或编码方式无关。md5()函数的格式如下：

```
string md5 ( string str [, bool raw_output] );
```

其中字符串 str 为要加密的明文，raw_output 参数如果设为真，则函数返回一个二进制形式的密文。raw_output 参数默认为 FALSE。

很多网站注册用户的密码都是先使用 MD5 加密，然后再保存到数据库中。用户登录时，程序是把用户输入的密码计算成 MD5 值，然后再去和数据库中保存的 MD5 值进行比较。在这个过程中，程序自身都不会“知道”用户的真实密码，从而保证了注册用户的个人隐私，提高了安全性。

19.1.3　使用 sha1()函数进行加密

和 MD5 类似的还有 SHA 算法。SHA 全称为 Secure Hash Algorithm（安全哈希算法），PHP 提供的 sha1()函数使用的就是 SHA，该函数的语法格式如下：

```
string sha1 ( string str [, bool raw_output] )
```

该函数返回一个 40 位的十六进制数，如果参数 raw_output 为真，则返回一个 20 位的二进制数。raw_output 默认为 FALSE。

注意： sha 后面的 1 是阿拉伯数字，不是字母 l（L），读者一定要注意。

19.2　PHP 加密扩展库

PHP 除了自带的几种加密函数外，还有功能更全面的加密扩展库 Mcrypt 和 Mhash。其中 Mcrypt 扩展库可以实现加密解密功能，就是说既能将明文加密，也可以将密文还原。Mhash 扩展库则包含了 MD5 在内的多种 hash 算法实现的混编函数。

19.2.1　Mcrypt 扩展库

1．Mcrypt 库安装

单项加密的优势是密文无法还原为明文，即使数据被截获也不会造成资料外泄。但有时，还是需要将密文还原成明文的，这时就需要使用双向加密技术。Mcrypt 是一个功能十分强大的加密算法扩展库。在标准的 PHP 安装过程中并没有把 Mcrypt 安装上，但 PHP 的主目录下包含了 libmcrypt.dll 和 libmhash.dll 文件（libmhash.dll 是 Mhash 扩展库，这里一起安装上），首先将文件复制到系统目录 windows\system32 下，然后在 php.ini 文件中找到 ;extension=php_mcrypt.dll ;extension=php_mhash.dll 这两句话，将前面的分号（;）去掉，最后重新启动服务器，就可以使用这两个扩展库了。

2．Mcrypt 库常量

Mcrypt 库支持 20 多种加密算法和 8 种加密模式。读者可以通过函数 mcrypt_list_algorithms()和 mcrypt_list_modes()来查看。

例如，通过函数 mcrypt_list_algorithms()和 mcrypt_list_modes()操作数据。代码如下：

```
<?php
    $en_dir = mcrypt_list_algorithms();                    //函数返回 Mcrypt 支持的加密算法数组
```



```
	echo "Mcrypt 支持的算法有：";
	foreach($en_dir as $en_value){
		echo $en_value." ";
}
?>
===============================
<?php
	$mo_dir = mcrypt_list_modes();						//函数返回 Mcrypt 支持的算法模式数组
	echo "<p>Mcrypt 支持的加密模式有：";
	foreach($mo_dir as $mo_value){
		echo $mo_value." ";
}
?>
```

运行结果如图 19.3 所示。

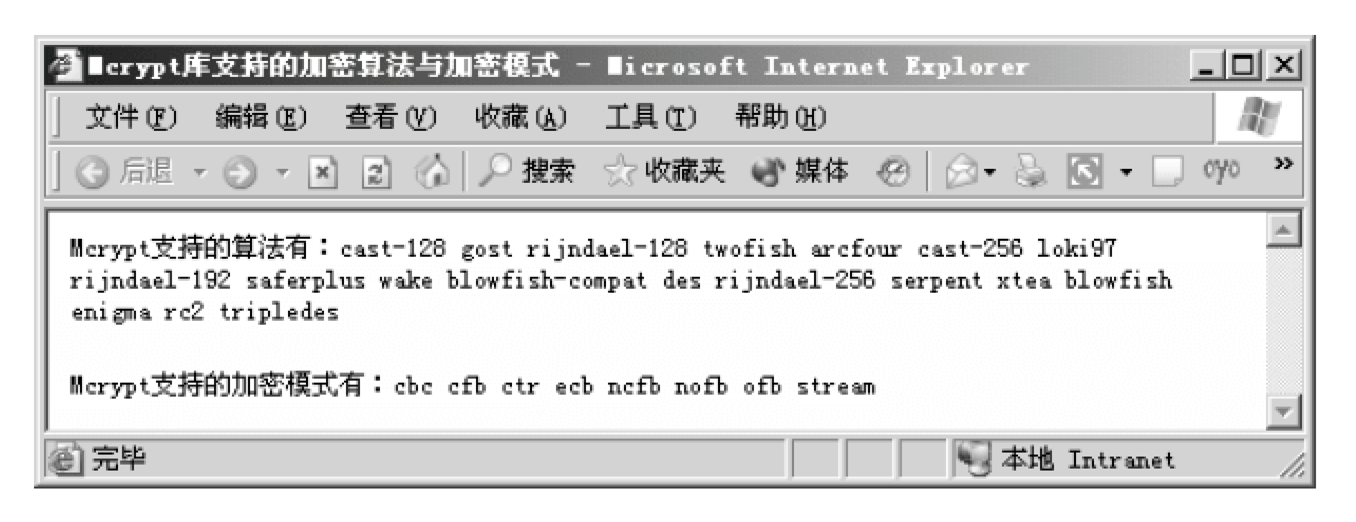


图 19.3　Mcrypt 库支持的加密算法与加密模式

这些算法和模式在实际应用中要用常量来表示。写的时候分别加上前缀“MCRYPT_”和“MCRYTP_MODE_”来表示，例如：

☑　twofish 算法表示为 MCRYPT_TWOFISH。

☑　cbc 加密模式表示为 MCRYPT_MODE_CBC。

19.2.2　Mhash 扩展库

1．Mhash 库安装

关于 Mhash 库的安装在前面已经介绍过了，这里不再重复。请读者参见 19.2.1 小节 Mcrypt 扩展库中的 Mcrypt 库安装。

2．Mhash 库常量

Mhash 库支持 MD5、SHA1、CRC32 等多种散列算法，可以使用 mhash_count()和 mhash_get_hash_name()函数输出支持的算法名称。

如果在实际应用中使用常量，需要在算法名称前面加上“MHASH_”前缀，如 CRC32 表示为 MHASH_CRC32。

3．Mhash 应用

相比 Mcrypt 扩展库的 30 多个函数，Mhash 库中只有 5 个函数，除了上面使用到的两个函数外，下面介绍其他的 3 个函数。

☑ mhash_get_block_size()函数

该函数的语法如下：

```
int mhash_get_block_size ( int hash )
```

该函数返回指定的 hash 所占字区大小，如 mhash_get_block_size(MHASH_CRC32)。

☑ mhash()函数

该函数的语法如下：

```
string mhash ( int hash, string data [, string key] )
```

该函数返回一个哈希值。参数 hash 为要使用的算法，参数 data 是要加密的数据，参数 key 是加密使用的密钥。

☑ mhash_keygen_s2k()函数

该函数的语法如下：

```
string mhash_keygen_s2k ( int hash, string password, string salt, int bytes )
```

该函数将根据参数 password 和 salt 返回一个长度为 bytes 的 key 值，参数 hash 为要使用的算法。其中 salt 为一个固定 8 字节的值，如果用户给出的数值小于 8bytes，将用 0 补齐。

19.3 PHP 漏洞和防护措施

服务器作为网站的载体其安全性是至关重要的。用户通过学习本节内容可以对常见漏洞进行防护，实现 PHP 的安全运行。

19.3.1 本地文件包含漏洞

在 PHP 的配置文件 php.ini 中存在一个参数 register_globals 用于设置是否允许用户注册全局变量。设置此参数可以为编程带来便利，但也存在一些安全隐患。

设置此参数的方法如下：

```
register_globals=Off
```

当将此参数设置为 On 的状态时可以允许 PHP 地址栏直接拼接参数。

例如，http://localhost/index.php?auth=1 演示漏洞的存在。代码如下：

```
<?php
	header("Content-Type:text/html;charset=utf-8");
	session_start();
?>
<form action="" method="post">
	<input type="text" name="text"/>
	<input type="submit" name="sub" value="确定"/>
</form>
<?php
	if($_POST[text]=="mr"){
		$auth=1;
	}
	if($auth==1){
		$_SESSION['user']="mr";
		echo "Succ";
	}
```

```
?>
```

用户只需要拼接参数，即可直接越过表单验证模块并输出成功登录的提示。

19.3.2　远程文件包含漏洞

PHP 对本地文件和远程文件进行的操作使用的是相同的函数。所以一些恶意用户通过强行手段利用网站的 PHP 代码包含并运行自己的脚本。

例如，编写一个 PHP 文件，首先获取超链接传递的参数值，然后在程序中通过 include 语句包含一个不存在的文件，那么在运行程序时就会返回一个错误信息。代码如下：

```
<?php
    $file=$_GET['name'];              //地址栏存在某个文件名称
    include($file);                   //包含该文件
?>
```

如果用户未对上述代码进行任何错误设置，在浏览器信息的显示上会报出如图 19.4 所示的错误信息。

由此非法用户就可以知道此部分操作是要包含一个文件，那么非法用户就可以通过拼接地址栏参数强行包含本地脚本文件并运行，如图 19.5 所示。



图 19.4　错误提示



图 19.5　包含漏洞

解决这种问题的方式常见的有两种方法：第一种方法完善错误信息，当包含文件出错时弹出提示信息，从而很好地隐藏错误信息，使用户不知道此部分的操作内容。第二种方法是使用正则表达式函数替换地址栏的一些操作符即可。

19.3.3　文件上传漏洞

文件上传在第 13 堂课中已进行了深入的讲解，关于漏洞的防护笔者也简单地介绍了一些。其实上传文件的防护操作是很简单的，主要是针对上传文件的文件格式进行限制。尽量避免那些可执行文件的上传操作；否则非法用户只需要一个简单的 HTML 文件，就可以使浏览器瞬间崩溃，这绝不是危言耸听。类似于使浏览器崩溃的 HTML 文件数不胜数。例如，PHP 的死循环文件就是一个典型的例子。死循环的代码如下：

```
<?php
   for($a=0;$a<100000000000000;$a++){
        for($b=1000000000000;$b>0;$b++){
             echo $a*$b;
        }
   }
?>
```

打开这样一个脚本文件需要耗费服务器的所有资源，最后使浏览器崩溃或者无响应。所以防护上传文件的漏洞只需规范化上传文件的格式即可。

19.3.4　提交表单漏洞

一般情况下，很多网站通常使用 JavaScript 脚本对表单操作进行验证，因为这样做的好处是 JavaScript 代码在客户端就可以执行，可以为服务器节省很多资源，但是这样的操作如果不善也会造成安全隐患。大家看下面的一段代码：

```
<?php
    header("Content-Type:text/html;charset=utf-8");                    //设置编码集
?>
<script language="javascript" src="jquery-1.4.2.js"></script>
<form name="form1" action="02.php" method="post">                     //表单
    <input type="text" name="text"/>
    <input type="submit" name="sub" value="确定"/>
</form>
<script>
    $(function(){                                                     //JavaScript 验证
        $("input:eq(1)").click(function(){
            var text=$("input:eq(0)").val();
            if(text!="mr"){
                alert("您不是权限用户");
                return false;
            }
        });
    })
</script>
```

显示效果如图 19.6 所示。



图 19.6　JavaScript 脚本限制表单提交

此段代码采用 JavaScript 脚本对表单登录进行验证，根据 JavaScript 脚本特性，非法用户可以将当前运行文件另存到本地静态 HTML 文件，删除 JavaScript 验证，将 action 跳转拼接到网站登录的成功页面，即可越过表单限制实现登录。修改后的脚本内容如下：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0023)http://localhost/01.php -->
<HTML><HEAD>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<FORM name=form1 action='http://localhost/02.php' method=post>
<META content="MSHTML 6.00.2900.5726" name=GENERATOR></HEAD>
<BODY><INPUT name=text> <INPUT type=submit value=确定  name=sub> </FORM>
</SCRIPT>
```

```
</BODY></HTML>
```

运行此文件，直接单击“确定”按钮，即可登录到成功页面，如图 19.7 所示。



图 19.7　越过表单验证

19.3.5　SQL 注入漏洞

SQL 注入是网络攻击的一种常用手法，攻击者通过对页面中的 SQL 语句进行拼接来获取管理员的账号和密码及更多的数据信息。此种攻击方式对网站的攻击最大。看下面一段代码：

```
<?php
    header("Content-Type:text/html;charset=utf-8");
     $conn=mysql_connect("localhost","root","111") or die("Connect MySQL False");
     mysql_select_db("db_database_test",$conn);
     mysql_query("SET NAMES utf8");
     $id=$_GET[id];
     $sql="select * from tb_admin where id= '$id'";
     $rs=mysql_query($sql);
     if(mysql_numrows($rs)){
            $row=mysql_fetch_row($rs);
            echo $row[0]."|".$row[1]."|".$row[2]."|";
     }else{
            echo "没有找到";
     }
?>
```

本代码通过获取 id 参数的值，构造一个用户自定义的 SQL 语句，执行查询操作。例如，从浏览器上访问 http://localhost/index.php?id=9 可以获得“没有找到”的结果。此时虽然用户没有得到结果，但是用户已经可以操作 SQL 语句了。就这样用户可以在地址栏中编写逻辑，只要逻辑正确就可以得到数据。本实例实现注入的地址栏代码为 http://localhost/index.php?id=9 and (select ascii(substr(name,1,1)) from tb_admin limit 0,1)=97，就可以正常显示网页数据，即 SQL 注入成功。

此代码防护只需简单的一行代码即可实现，在获取 id 参数值时进行数据类型转换并过滤掉非法的 SQL 字符。代码如下：

```
$id=(int)$_GET[id];
```

19.4　照猫画虎——基本功训练

19.4.1　基本功训练 1——通过 crypt()函数进行加密

视频讲解：光盘\mr\19\lx\通过 crypt()函数进行加密.exe
实例位置：光盘\mr\19\zmhh\01

crypt()加密函数在实际的 PHP 开发过程中是比较常见的，其实此函数与 md5()、sha1()函数在使用上并

无区别，也不存在哪个加密函数更优越。因为它们的加密操作都是不可逆的。本训练采用 crypt()函数实现加密操作，运行结果如图 19.8 所示。



图 19.8　crypt()函数实现加密

实现本训练的原理非常简单，首先定义一个需要加密的字符串，然后通过 crypt()函数实现加密操作。核心代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");      //设置编码集
    $str = 'hello world!';
    echo '数据加密前：'.$str."<br>";
    $crypttostr = crypt($str);                            //加密操作
    echo '数据加密后：'.$crypttostr;
?>
```

照猫画虎：实现对特殊字符的加密。（20 分）（实例位置：光盘\mr\19\zmhh\01_zmhh）

19.4.2　基本功训练 2——通过 md5()函数进行加密

视频讲解：光盘\mr\19\lx\通过 md5()函数进行加密.exe

实例位置：光盘\mr\19\zmhh\02

md5()加密函数是笔者比较喜欢的加密函数，因为经过此函数加密后返回的数据较其他函数更整洁，给人的直观感觉不像乱码。本实例的运行结果如图 19.9 所示。



图 19.9　md5()实现加密

实现本实例的原理同样非常简单，操作方法与 19.4.1 小节相同，这里不再赘述。代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");
    $str = 'hello world!';
    echo '数据加密前：'.$str."<br>";
    $md5str = md5($str);
    echo '数据加密后：'.$md5str;
?>
```

照猫画虎：利用 md5()函数实现对中文字符串的加密。（20 分）（实例位置：光盘\mr\19\zmhh\02_zmhh）

19.4.3　基本功训练 3——通过 sha1()函数进行加密

视频讲解：光盘\mr\19\lx\通过 sha1()函数进行加密.exe

实例位置：光盘\mr\19\zmhh\03

sha1()函数加密后的字符串格式与 md5()十分近似，也是比较常用的一种加密格式，其加密后的效果如图 19.10 所示。

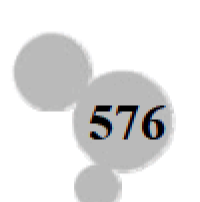




图 19.10　sha1()函数加密

代码如下：

```
<?php
    header("Content-Type:text/html;charset=utf-8");
    $str = 'hello world!';
    echo '数据加密前：'.$str."<br>";
    $md5str = sha1($str);
    echo '数据加密后：'.$md5str;
?>
```

照猫画虎：利用 sha1()函数分别加密“PHP”和“php”字符串，观察结果是否相同。(20 分)(实例位置：光盘\mr\19\zmhh\03_zmhh)

19.4.4　基本功训练 4——通过 mcrypt()函数进行加密、解密操作

视频讲解：光盘\mr\19\lx\通过 mcrypt()函数进行加密、解密操作.exe

实例位置：光盘\mr\19\zmhh\04

使用 mcrypt()函数实现加密操作虽然比较稳妥，但是由于操作起来相对较麻烦，并不十分常用。但也是一种十分优秀的加密方法。本实例的运行结果如图 19.11 所示。

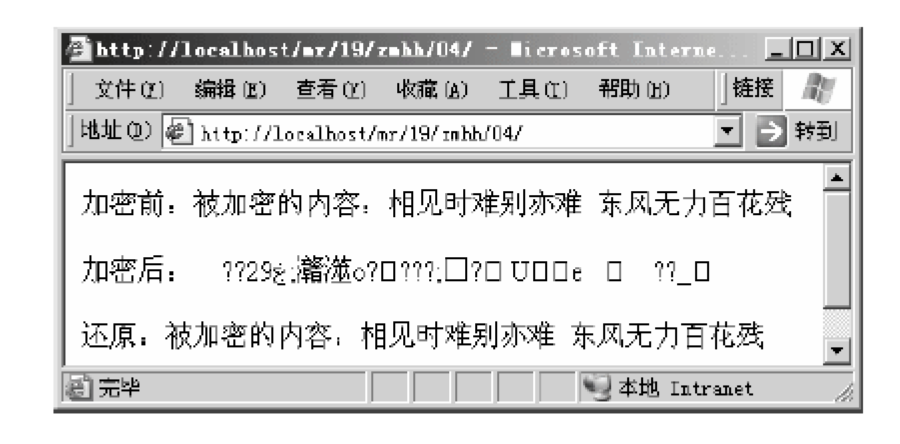


图 19.11　mcrypt()函数加密

通过 mcrypt_create_iv()将指定的字符串加密，然后利用 mcrypt_decrypt()函数进行解密。代码如下：

```
<?php
    $str = "被加密的内容：相见时难别亦难  东风无力百花残";                    //加密文本
    $key = "key:111";                                                         //密钥
    $cipher = MCRYPT_DES;                                                     //密码类型
    $modes = MCRYPT_MODE_ECB;                                                 //密码模式
    $iv = mcrypt_create_iv(mcrypt_get_iv_size($cipher,$modes),MCRYPT_RAND);   //初始化向量
    echo "加密前：".$str."<p>";
    //加密：
    $str_encrypt = mcrypt_encrypt($cipher,$key,$str,$modes,$iv);              //加密函数
    echo "加密后：".$str_encrypt." <p>";
    $str_decrypt = mcrypt_decrypt($cipher,$key,$str_encrypt,$modes,$iv);      //解密函数
    echo "还原：".$str_decrypt."<p>";
?>
```

照猫画虎：利用 mcrypt()函数实现对英文字符串的加密。（20 分）（实例位置：光盘\mr\19\zmhh\04_zmhh）

19.4.5　基本功训练 5——通过 mhash()扩展库进行加密、解密

视频讲解：光盘\mr\19\lx\通过 mhash()扩展库进行加密、解密.exe
实例位置：光盘\mr\19\zmhh\05

使用 mhash()扩展库实现加密操作相对较复杂，需要开启 PHP 的模块支持，所以实际应用相对较少。本实例的运行结果如图 19.12 所示。



图 19.12　mhash()扩展库实现加密

代码如下：

```
<?php
    $filename = 'index.txt';                                    //文件路径
    $str = file_get_contents($filename);                        //读取文件内容到变量$str 中
    $hash = 2;                                                  //设置 hash 值
    $password = '111';                                          //设置变量$password
    $salt = '1234';                                             //设置变量$salt
    $key = mhash_keygen_s2k(1,$password,$salt,10);              //生成 key 值
    $str_mhash = bin2hex(mhash($hash,$str,$key));               //使用$key 值、$hash 值对字串$str 加密
    echo "文件 index.txt 的校验码是：".$str_mhash;              //输出校验码
?>
```

照猫画虎：利用 mhash()函数实现对数字的加密。（20 分）（实例位置：光盘\mr\19\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

19.5　情景应用——拓展与实践

19.5.1　情景应用 1——通过 crypt()函数对用户注册密码进行加密

视频讲解：光盘\mr\19\lx\通过 crypt()函数对用户注册密码进行加密.exe
实例位置：光盘\mr\19\qjyy\01

加密操作一般都被应用在对用户的用户名、密码的加密操作中。如果是普通的门户网站，网站管理员可能不用做太多的细节。但是像淘宝这样的电子商务网站，加密操作就显得尤为重要了。如果不对用户的各种操作进行加密包括对支付宝的加密，非法用户一旦获取用户名和密码甚至是银行的账号密码，那么结果是不可想象的。本实例采用 crypt()函数对用户的密码进行加密，运行结果如图 19.13 所示。



图 19.13　实现对密码加密

本实例的原理非常简单，当用户单击“登录”按钮时，动态获取用户的密码，然后通过 crypt()函数实现加密操作。代码如下：

```
	<?php
	if(isset($_POST[sub])){                                  //当“登录”按钮被单击时
		if($_POST[text]!="" and $_POST[pwd]!=""){            //判断文本框是否为空
			echo "密码加密值为：".crypt($_POST[pwd]);        //输出加密后的密码
		}else{
			echo "文本框不能为空";                           //输出提示
		}
	}
?>
```

DIY：利用 crypt()函数实现对用户名和密码的双重加密。（20 分）（实例位置：光盘\mr\19\qjyy\01_diy）

19.5.2　情景应用 2——通过 md5()函数对用户注册密码进行加密

视频讲解：光盘\mr\19\lx\通过 md5()函数对用户注册密码进行加密.exe

实例位置：光盘\mr\19\qjyy\02

几乎所有的加密操作都是针对用户的用户名和密码而进行的，md5 加密函数也不例外。本实例通过 md5()函数对密码进行加密，运行结果如图 19.14 所示。

图 19.14　实现对密码加密

代码如下：

```
<?php
	if(isset($_POST[sub])){
		if($_POST[text]!="" and $_POST[pwd]!=""){
			echo "密码加密值为：".md5($_POST[pwd]);
		}else{
			echo "文本框不能为空";
		}
	}
?>
```

DIY：利用 md5()函数实现对用户名和密码的双重加密。（20 分）（实例位置：光盘\mr\19\qjyy\02_diy）

19.5.3 情景应用 3——通过自定义函数实现加密操作

视频讲解：光盘\mr\19\lx\通过自定义函数实现加密操作.exe

实例位置：光盘\mr\19\qjyy\03

截止到发稿前，笔者对所有的加密函数进行了一些调查，网络上已经对相关的不可逆加密函数进行了解析，其实所谓的不可逆加密函数在网络上已经变为可逆，以前大家所信赖的这些加密函数已经不再安全了，但是没关系，用户可以在加密之前运用一些小的技巧，混淆非法用户的视听。本实例封装一个自定义函数，通过自定义函数对数据进行加密，其效果如图 19.15 所示。



图 19.15 自定义函数实现加密

原理同样是通过 md5 的加密手段实现的，不同的是笔者拼接一个字符串，就算非法用户破解了 md5 加密的密码，但是仍然得不到真正的密码。代码如下：

```
<?php
	function jm($str){				//自定义函数
		$string=$str."PHP";			//拼接字符串
		$string=md5($string);		//加密操作
		echo $string;				//输出
	}
	if(isset($_POST[sub])){
		if($_POST[text]!=""){
			jm($_POST[pwd]);
		}else{
			echo "文本框不能为空";
		}
	}
?>
```

DIY：封装一个加密类。（20 分）（实例位置：光盘\mr\19\qjyy\03_diy）

19.5.4 情景应用 4——隐藏 PHP 文件扩展名

视频讲解：光盘\mr\19\lx\隐藏 PHP 文件扩展名.exe

实例位置：光盘\mr\19\qjyy\04

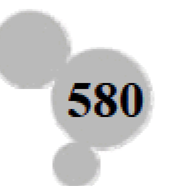
为提高网站的安全性，可以使用隐藏 PHP 的方法，虽然这种方法对提高安全性的作用不大，但是在某些情况下，尽可能地多增加一份安全性还是值得的。在本实例中实现对 PHP 的隐藏，使其看上去像其他的编程语言。本实例的运行结果如图 19.16 所示。

在运行本实例时，IE 地址栏中显示的文件后缀是.asp，这就是隐藏 PHP 的结果，使其看上去似乎是使用 ASP 语言开发的。同样在查看程序的源文件时，看到使用的文件仍然是以.asp 为后缀，如图 19.17 所示。

隐藏 PHP 文件扩展名的步骤如下：

（1）对 Apache 服务器中的配置文件 httpd.conf 进行配置，首先定位到如下位置：

```
<IfModule dir_module>
    DirectoryIndex index.php index.asp index.html index.htm
</IfModule>
```



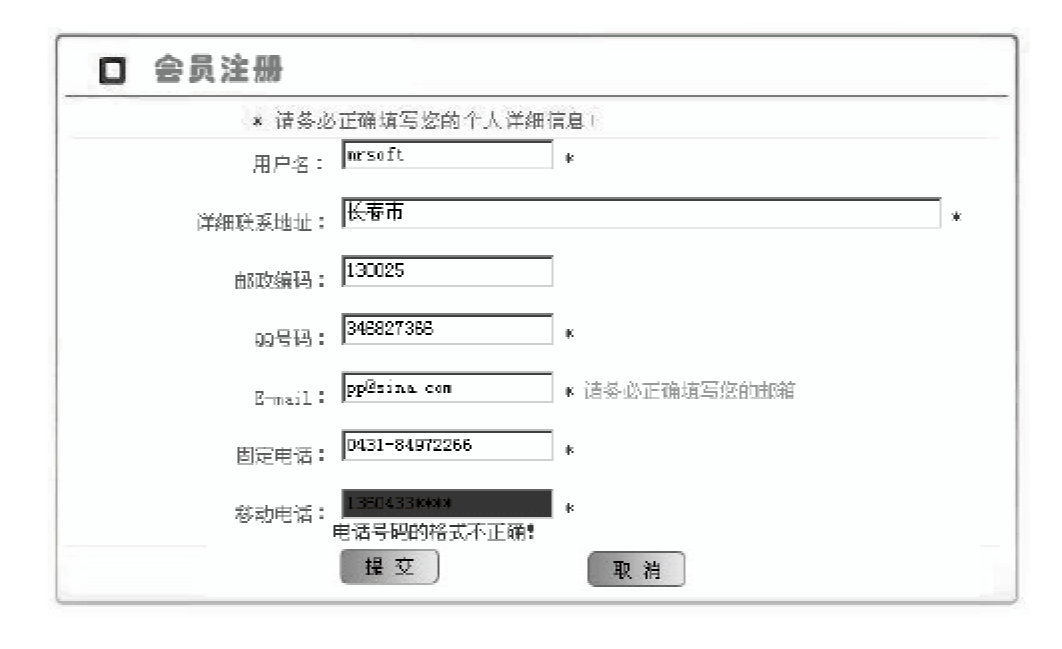


图 19.16　隐藏 PHP 文件的扩展名

```
index[1] - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
    <td> </td>
    <td align="center" valign="top">
        <form name="form_reg" method="post"
action="index_ok.asp" onSubmit="return chkreginfo
(form_reg,'all')">
      <table width="620" height="262" border="0"
align="center" cellpadding="0" cellspacing="0">
```

图 19.17　查看源文件的结果

增加一个扩展名 index.asp。

（2）然后再定位到如下位置：

```
<IfModule mod_php5.c>
    #指定 PHP 文件的扩展名
    AddType application/x-httpd-php .asp .jsp .net
    #AddType application/x-httpd-php .php
    #AddType application/x-httpd-php .php3
    #AddType application/x-httpd-php-source .phps
</IfModule>
```

指定 PHP 文件的扩展名为.asp、.jsp、.net。

（3）保存文件，并重新启动 Apache 服务器。

（4）创建 index.php 文件，完成用户注册表单的生成，并通过 JavaScript 脚本对表单中的数据进行无刷新验证，将表单中的数据提交到 index_ok.asp 文件中。创建完成后将该文件的后缀修改为.asp。

（5）创建 index_ok.php 文件，通过 POST 全局变量获取表单中提交的数据，并进行输出。创建完成后将该文件的后缀修改为.asp。

DIY：通过伪静态在论坛中隐藏 PHP。（20 分）（实例位置：光盘\mr\19\qjyy\04_diy）

19.5.5　情景应用 5——禁止用户输入敏感字符

视频讲解：光盘\mr\19\lx\禁止用户输入敏感字符.exe

实例位置：光盘\mr\19\qjyy\05

在论坛、博客或者社区等这种具有言论自由的网站中，在发布评论时偶尔会出现如图 19.18 所示的提示信息。

这是为了保证网站内容健康、积极、向上，不传播非法内容而设置的一个功能，当有人在网站上散布非法内容时，网站本身就给出提示信息，不允许传播非法内容。这也就是本实例中将要讲解的内容，如何实现禁止用户输入敏感字符，主要应用 substr_count()函数、count()函数、for 循环语句和 if 条件语句，以及一个自定义数组。



图 19.18　内容中含有敏感词

（1）substr_count()函数用来获取指定字符在字符串中出现的次数。

```
int substr_count(string haystack,string needle)
```

参数 haystack 是指定的字符串，参数 needle 为指定的字符。

（2）自定义数组$keyword，该数组中存储的是敏感关键字。本实例中将明日科技、编程词典、明日图

书、明日软件作为敏感词。

```
$keyword = array ('明日科技', '编程词典', '明日图书', '明日软件' );
```

（3）count()函数用来统计数组中元素的个数。

（4）根据数组中元素的个数，执行 for 语句，循环输出数组中元素的值。

（5）在 if 语句中根据 substr_count()函数的返回值判断数据中是否含有关键字。

具体实现步骤如下：

（1）创建 index.php 页面，连接数据库，输出数据表中存储的发布帖子内容；创建表单，添加表单元素，完成帖子标题和内容的提交。关键代码如下：

```
<?php	include "conn/conn.php";  ?>
<table width="754" border="0" align="center" cellpadding="0" cellspacing="1">
<?php
	include("function.php");		//调用 unhtml 函数，用于过滤帖子内容
	$sql=mysql_query("select * from tb_guestbook order by createtime desc");
	$info=mysql_fetch_array($sql); //获取查询结果
	if($info==false)
	{
		echo "暂无帖子";
	}else{
		do{				//通过 do…while 循环输出帖子内容，并对帖子内容进行过滤
?>
	<tr>
		<td width="151" height="25" bgcolor="#FFFFFF"><div align="center">主   题：
</div></td>
		<td width="600" bgcolor="#FFFFFF"> <?php echo unhtml($info[title]);?></td>
	</tr>
	<tr>
		<td height="95" bgcolor="#FFFFFF"><div align="center">内   容：</div></td>
		<td bgcolor="#FFFFFF"> <?php echo unhtml($info[content]);?></td>
	</tr>
	<?php
		}
		 while($info=mysql_fetch_array($sql));
	}
	?>
<table width="754" height="150" border="0" align="center" cellpadding="0" cellspacing="1">
	<form name="form1" method="post" action="index_ok.php" onSubmit="return chkinput(this)">
	<tr>
		<td height="20" colspan="2" bgcolor="#6090BE"><div align="center" class="STYLE1">发布帖子
</div></td>
	</tr>
	<tr>
		<td width="149" height="25" bgcolor="#FFFFFF"><div align="center">主   题：
</div></td>
		<td width="602" bgcolor="#FFFFFF"> <input type="text" name="title" size="35" class="inputcss">
</td>
	</tr>
	<tr>
		<td height="100" bgcolor="#FFFFFF"><div align="center">内   容：</div></td>
		<td height="100" bgcolor="#FFFFFF"> <textarea name="content" cols="60" rows="6"
```

```
class="inputcss"></textarea></td>
        </tr>
        <tr>
                <td height="25" colspan="2" bgcolor="#FFFFFF"><div align="center"><input type="submit" value="提交" class="buttoncss">  <input type="reset" value="重置" class="buttoncss"></div></td>
        </tr>
        </form>
</table>
```

（2）创建 index_ok.php 文件，获取表单中提交的数据，检测提交的数据中是否包含指定的敏感词，如果包含敏感词则给出提示信息，并跳转到帖子发布页面，否则将数据添加到指定的数据表中。代码如下：

```
<?php
include "conn/conn.php";                    //连接数据库
$title = $_POST [title];                    //获取帖子标题
$content = $_POST [content];                //获取帖子内容
$createtime = date ( "Y-m-j H:i:s" );       //获取当前时间
$keyword = array ('明日科技', '编程词典', '明日图书', '明日软件' );      //定义敏感词
$m = 0;
for($i = 0; $i < count ( $keyword ); $i ++) { //根据数组元素数量执行 for 循环
        //应用 substr_count 检测帖子的标题和内容中是否包含敏感词
        if (substr_count ( $title, $keywod [$i] ) > 0 or substr_count ( $content, $keyword [$i] ) > 0) {
                $m ++;
        }
}
if ($m > 0) {                               //如果变量 m 的值大于 0 说明包含敏感词
        echo "<script>alert('您输入的内容中含有敏感关键字'); history.back();</script>";
} else {
        //否则说明不包含敏感词，则将数据添加到数据表中
        if (mysql_query ( "insert into tb_guestbook(title,content,createtime) values ('$title','$content','$createtime')" )) {
                echo "<script>alert('帖子发表成功!');window.location.href='index.php';</script>";
        } else {
                echo "<script>alert('帖子发表失败!');history.back();</script>";
        }
}
?>
```

DIY：检测用户注册的用户名是否包含非法关键字。（20 分）（实例位置：光盘\mr\19\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

19.6　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．下面（　　）加密函数是可逆的。

A．md5　　　　B．sha1　　　　C．mcrypt

2．md5 加密函数把不同长度的数据信息最终加密为（　　）位数值。

A．64　　B．32　　C．128　　D．256

3．sha1 函数返回 40 位（　　）进制数。

A．八　　B．十六　　C．二

4．在标准的 PHP 安装过程中并没有把 Mcrypt 安装上，但 PHP 的主目录下包含了 libmcrypt.dll 和 libmhash.dll 文件（libmhash.dll 是 Mhash 扩展库，需要把此扩展库复制到（　　）下）。

A．system32　　B．system　　C．windows

5．用户注册全局变量需要修改 php.ini 文件中的（　　）。

A．register_globals　　B．y2k_compliance　　C．output_buffering

二、填空题（每题 10 分，5 道题）

1．利用 crypt()函数实现将“PHP”加密，请将代码补充完整。

```
$string = "PHP";
$string = （                    ）
echo $string;
```

2．利用 md5()函数实现将“PHP”加密，请将代码补充完整。

```
$string = "PHP";
$string = （                    ）
echo $string;
```

3．利用 sha1()函数实现将“PHP”加密，请将代码补充完整。

```
$string = "PHP";
$string = （                    ）
echo $string;
```

4．请填写防止上传文件漏洞的方法（　　　　）。

5．请填写如何防止 JavaScript 脚本越过表单提交的限制方法（　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

19.7　行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。

续表

序号	内　　容	行 动 指 南	
1	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）利用 md5()函数加密控制用户登录权限。	
		（2）将加密后的字符串存储于数据库。	
		（3）将加密后的字符串存储于文件并实现读取。	
		（4）制作一个安全的用户留言板系统。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

19.8　成功可以复制——初中站长的创业故事李兴平

成就：“hao123”之父，创建国内著名个人网站 www.hao123.com，网站流量排名全球第 15 名。全球流量最大的中文网站之一。

创始公司：个人

财富巅峰：千万富翁，并享有部分百度公司股份。

最高学历：高中

出生日期：1979 年 6 月

超级语录：我天天泡在网吧里，我知道他们喜欢哪样的网站，讨厌哪些东西。我首先是一个互联网的热爱者，其次才是一个商人。

成功历程

1979 年，李兴平出生于广东兴宁县一个农民家庭。1999 年，网络在中国逐渐普及，兴宁县也开始有了网吧，李兴平开始迷上了上网。初中毕业后，在当地网吧找了一份网吧管理员的工作。因为要帮人攒电脑赚钱，他需要用网络查询配件报价之类的信息，但很快发觉在网上找资料非常困难，当时的中文网站不仅内容不够丰富，数量有限，而且要把那些用英文字母表示的网址一个个记下来，并不是一件容易的事情。他利用平时学到的网页制作知识设计了一个个人网页，把他认为好的网站搜集在一起，并和它们建立链接。

后来，李兴平的电脑知识越来越丰富，也能设计出更复杂的网页，但是，他发现其实静态页面简单又快捷，对于个人网站，比动态页要适用。

为了让网页方便使用，他把当时中国排名前 5000 位的站点进行分门别类，做了一个“网址大全”式的导航网页，并赋予个“其貌不扬”的名字“hao123”。因为网站一直只有李兴平一个人在管理，他的工作量也很大，一天大概有 6~7 个小时是在检查、更新链接。但链接太多，不可能每天都全部检查，因此他经常担心某些网站含有非法链接，压力和工作量都很大。功夫不负有心人，因简单、易用、精确，hao123 网站逐渐被广大网民所认识，并被大部分普通网民设为首页。2003 年，hao123 的网址导航站点已经占领了中国网吧 50%以上的首页，这简陋得让精英们不屑一顾的 hao123.com，也慢慢成为全球流量最大的中文网站之一。



hao123.com 主页

为了让网站能有些收入，他开始做一些点击广告以及网站联盟等。随着流量增大，网站有了固定收入。最初的广告费一个月只有几百元，后来逐渐增加到 1000 元，4000 元。到 2001 年，一天竟能达到了几万元。2003 年后，每个月的收入稳定在 80 万元左右。

现在，hao123 已成为中国网民上网的第一门户和互联网网址导航的第一品牌。李兴平也成为中国网络界当之无愧的传奇人物。

深度评价

李兴平的成功也许会让那些怀揣着宏大的“商业计划书”以及数千万美元的回国创业的海龟们愤愤不平，凭什么这样一个既不懂技术，又不懂管理，不会讲洋文更没有强大资金背景支持的“土八路”赢得了这场战争？hao123 的成功真正说明，技术并不是最重要的，重要的是你是否能“幸福着用户的幸福，痛苦着用户的痛苦”、能否对所做的事业极其无限热爱和超级专注。

第20堂课

PHP 的字符编码

（视频讲解：79 分钟）

网页字符编码就是计算机在显示字符时使用的编码，也就是一个字符集。在 PHP 中，为网页设置准确、适合的字符编码是非常重要的，只有使用正确的字符编码，网页才能够流畅、清晰地输出数据，否则网页中将会输出乱码 。本堂课将详细地讲解如何在 PHP 中设置字符编码，以及出现字符编码问题的解决方法。

学习摘要：

- **字符集和编码**
- **PHP 网页的字符编码**
- **PHP 开发中的乱码问题**

20.1 字符集和编码

字符是各种文字和符号的总称，包括各个国家的文字、数字、标点符号和图形符号等。字符集和编码经常出现在一起，但它们实际上一个是名词，一个是动词。字符集是将人类使用的自然文字映射到计算机内部二进制的表示方法，是某种文字和字符的集合，作为名词使用，如 GB2312 字符集、ISO 8859 字符集；而编码是对这种字符集的编码方式，作为动词使用。

计算机中的“字符”都是用一个二进制数字来表示的，这个二进制数字称为字符的编码，用于不同国家的语言在计算机中的存储和解释规范，而针对不同需求选择的字符集合，被定义为不同的字符集，如GB2312。

中文文字数目大，而且还分为简体中文和繁体中文两种不同书写规则的文字，而计算机最初是按英语单字节字符设计的，因此，对中文字符进行编码，是中文信息交流的技术基础。

20.1.1 ISO 8859 字符集

ISO 8859 是由 ISO（International Organization for Standardization，国际标准化组织）在 ASCII 编码基础上制定的编码标准。ISO 8859 包括 128 个 ASCII 字符，并新增加了 128 个字符，用于西欧国家的符号。

ISO 8859 存在不同的语言家族：

☑ Latin-1（西欧语言）。
☑ Latin-2（非 Cyrillic 的中欧和东欧语言）。
☑ Latin-3（南欧语言）。
☑ Latin-4（土耳其语言）。
☑ Latin-5（北欧和波罗地语言）。
☑ Latin-6（Cyrillic 西里尔语言）。
☑ Latin-7（阿拉伯语言）。
☑ Latin-8（希伯来语言）。

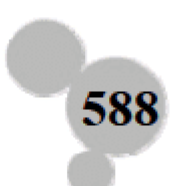
20.1.2 GB2312 与 GBK 字符集

GB2312 又称为 GB2312-80 编码，全称为《信息交换用汉字编码字符集·基本集》，由原中国国家标准总局发布，1981 年 5 月 1 日实施。GB2312 编码是中华人民共和国国家标准汉字信息交换用编码。它所收录的汉字已经覆盖 99.75%的使用频率，基本满足了汉字的计算机处理需要。在中国大陆和新加坡获广泛使用。

GB2312 编码收录简化汉字及一般符号、序号、数字、拉丁字母、日文假名、希腊字母、俄文字母、汉语拼音符号、汉语注音字母，共 7445 个图形字符。其中包括 6763 个汉字，其中一级汉字 3755 个，二级汉字 3008 个；包括拉丁字母、希腊字母、日文平假名及片假名字母、俄语西里尔字母在内的 682 个全角字符。

GB2312 编码中对所收汉字进行了“分区”处理，每区含有 94 个汉字/符号。这种表示方式也称为区位码。

各区包含的字符如下：01～09 区为特殊符号；16～55 区为一级汉字，按拼音排序；56～87 区为二级汉



字，按部首/笔画排序；10～15 区及 88～94 区则未有编码。

两个字节中前面的字节为第一字节，后面的字节为第二字节。习惯上称第一字节为“高字节”，而称第二字节为“低字节”。“高位字节”使用了 0xA1～0xF7（把 01～87 区的区号加上 0xA0），“低位字节”使用了 0xA1～0xFE（把 01～94 加上 0xA0）。

以 GB2312 编码的第一个汉字“啊”字为例，它的区号 16，位号 01，则区位码是 1601，在大多数计算机程序中，高字节和低字节分别加 0xA0 得到程序的汉字处理编码 0xB0A1。计算公式是：0xB0=0xA0+16，0xA1=0xA0+1。

GBK 编码是对 GB2312 编码的扩充，它包含 2 万多个字符，除了保持和 GB2312 兼容外，还包含并且增加了部分 Unicode 中没有的字符。GBK 只是一个规范而不是国家标准的编码，新的国家标准是 GB18030-2000，它比 GBK 包含更多的字符。

20.1.3　Unicode 字符集

Unicode 字符集是 Universal Multiple-Octet Coded Character Set 通用多 8 位编码字符集的简称，是由一个名为 Unicode 学术学会（Unicode Consortium）的机构制定的字符编码系统，支持现今世界各种不同语言的书面文本的交换、处理及显示。该编码于 1990 年开始研发，1994 年正式公布，最新版本是 2005 年 3 月 31 日的 Unicode 4.1.0。

Unicode 是一种在计算机上使用的字符编码。它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换和处理的要求。

Unicode 标准始终使用十六进制数字，而且书写时在前面加上前缀“U+”，例如字母“A”的编码为 004116 和字符“□”的编码为 20AC16。所以“A”的编码书写为“U+0041”。

Unicode 堪称一个伟大的成就，它把在这个世界上的每一个合理的文字系统整合成了一个单一的字符集。

20.1.4　UTF-8 编码

UTF-8 编码是 Unicode 中的一个使用方式。UTF 是 Unicode Translation Format，即把 Unicode 转做某种格式的意思。

UTF-8 便于不同的计算机之间使用网络传输不同语言和编码的文字，使得双字节的 Unicode 能够在现存的处理单字节的系统上正确传输。

UTF-8 使用可变长度字节来储存 Unicode 字符，例如 ASCII 字母继续使用 1 字节储存，重音文字、希腊字母或西里尔字母等使用 2 字节来储存，而常用的汉字就要使用 3 字节。辅助平面字符则使用 4 字节。

UTF-8 有两大优点：

（1）128 以下编码和单字节处理软件兼容。

（2）UTF-8 的多字节编码没有部分字节混淆问题。例如，删除半汉字后整行乱码的问题在 UTF-8 中是不会出现的；任何一个字节的损坏都只影响对应的那个字符，其他字符都可以完整恢复。

注意： *在开发项目的过程中，建议使用 UTF-8 编码，因为 UTF-8 在不同的计算机之间使用网络传输不同语言和编码的文字时，不会出现乱码的问题。*

20.2　PHP 网页的字符编码

在通过 PHP 语言开发 Web 项目时，对网页编码格式的选择非常重要，它关系着项目是否能够接受各种不同环境的考验。例如，项目编码格式采用 GB2312，那么当项目被修改为繁体中文编码时，就会出现乱码的问题，而如果最初使用 UTF-8 编码，那么无论做怎样的修改，项目的输出都不会出现乱码的问题。

20.2.1　设置编码格式

在 HTML/XHTML 页面的 head 区域中，Meta 是重要标签之一。其中，Meta 的 HTTP-EQUIV 关键字用于描述网页使用的字符集。常用的 HTTP-EQUIV 类型有 Content-Type 和 Content-Language（显示字符集的设定）。

这两个标签说明网页使用的语言，浏览器会据此来调用相应的字符集显示内容。例如：

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

meta 标签是比较重要的，它指定该 HTML 页面所使用的字符集为 UTF-8，浏览器会按它指定的编码显示，包括表单提交时的字符集编码。

Content-Type 的 content 值除了可以是“text/html”，还可以声明为“text/xml”，表示该文件是 XML 文档。

如果其中的“charset=utf-8”替换成“charset=BIG5”，则表示页面所用的字符集就是繁体中文 BIG5 码，首先应该确定该网页确实是 BIG5 编码，否则将出现乱码。

常用的 Charset 有 ISO-8859-1（英文）、BIG5（中国台湾地区使用的繁体中文）、utf-8、Shift-JIS（日语）、x-eucjp（日语）、euc-kr/koi8-r（韩语）、US-ASCII（ASCII 基本字符集），GB2312/EUC-CN、GBK/CP936 等字符集设置。

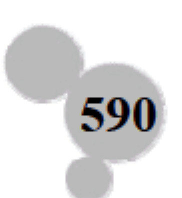
另外，在 HTML 页中，meta http-equiv= “Content-Language”表示网页使用的语言，它的 Content 声明可以是 EN、FR、zh_CN、zh_TW 等语言代码，用来告诉搜索引擎利用此标签分类文件语言。例如，告知搜索引擎文件使用的是中文的代码如下：

```
<meta http-equiv="Content-Language" content="zh_CN">
```

在 HTML 中，通过 meta 标签设置页面的编码格式，代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>无标题文档</title>
</head>
<body>
</body>
</html>
```

文件编码指的是文件（.html、.php）本身以何种编码形式保存。在不同的编辑工具中默认创建的文件编码格式是不同的。



20.2.2　转换编码格式

在 PHP 中，如果函数返回值的编码与页面设置的编码不同，那么就需要对函数返回值的编码格式进行转换。转换编码可以通过 iconv()函数和 mb_convert_encoding()函数来实现。

1．通过 iconv()函数实现编码转换

iconv()函数，转换被指定字符串的编码格式。其语法如下：

```
iconv ( string in_charset, string out_charset, string str )
```

该函数将指定的字符串 str 由 in_charset 编码格式转换成 out_charset 编码格式，成功返回转换后的字符串；失败返回 FALSE。

如果在参数 out_charset 后添加“//IGNORE”参数，那么将忽略转换时的错误。如果在出现错误时没有使用//IGNORE 参数，那么将导致出错字符串后面的所有字符串无法保存。

用 iconv()函数实现 GB2312 到 UTF-8 的编码转换，代码如下：

```
iconv("GB2312","UTF-8",$text);
```

用 iconv()函数实现 UTF-8 到 GB2312 的编码转换，代码如下：

```
iconv("UTF-8","GB2312",$text);
```

iconv()函数的另一种格式是使用“//IGNORE”参数，代码如下：

```
iconv("UTF-8","GB2312//IGNORE ",$text);
```

IGNORE 的作用是忽略转换时的错误，有时 iconv()函数在转换字符“—”到 GB2312 编码时会出错，如果没有 IGNORE 参数，该字符后面的所有字符串都无法被保存。

2．通过 mb_convert_encoding()函数实现编码转换

mb_convert_encoding()函数实现字符编码的转换。其语法如下：

```
mb_convert_encoding ( string str, string to_encoding [, mixed from_encoding] )
```

mb_convert_encoding()函数将字符串 str 的编码由 from_encoding 转换为 to_encoding。如果指定多个 from_encoding 编码，则使用逗号进行分隔；如果没有指定 from_encoding 编码，则使用内在的编码。

mb_convert_encoding()函数是 mbstring 扩展库中的函数，相对于 iconv()函数而言，mb_convert_encoding()函数的执行速度更快。例如：

用 mb_convert_encoding()函数实现 GB2312 到 UTF-8 的编码转换，代码如下：

```
mb_convert_encoding($text,"UTF-8","GB2312");
```

注意：由于 mb_convert_encoding()函数属于 mbstring 扩展库，所以要应用 mb_convert_encoding()函数，必须加载 mbstring 扩展库。即在 php.ini 文件中定位到“;php_mbstring.dll”，将其前面的分号“;”去掉，保存并重新启动 Apache 服务器。

例 20.01　为了更好地理解和应用 PHP 中的编码转换函数，下面编写一个实例，分别应用 iconv()函数和 mb_convert_encoding()函数，实现由 GB2312 编码到 UTF8 编码的转换。具体步骤如下。（**实例位置：光盘\mr\20\sl\20.01**）

（1）通过 Dreamweaver 新建一个动态 PHP 文件，设置网页默认编码格式为 GB2312。

（2）编写 PHP 脚本，定义两个字符串变量，并且通过 iconv()函数和 mb_convert_encoding()函数完成对

字符串由 GB2312 编码到 UTF8 编码格式的转换。代码如下：

```
<?php
$texts="明日科技";                                          //定义字符串
$strs=iconv("gb2312","utf-8",$texts);                      //将 GB2312 编码转换成 UTF8 编码
echo $strs;                                                //输出字符串
echo "<br>";
$text="编程词典";                                           //定义字符串
$str=mb_convert_encoding($text,"UTF8","gb2312"); //将 GB2312 编码转换成 UTF8 编码
echo $str;
?>
```

运行本实例，由于在 GB2312 编码格式的页面中输出 UTF8 编码的字符串，所以导致输出字符串乱码，如图 20.1 所示。

此时，在 IE 浏览器中选择“查看”/“编码”/Unicode（UTF-8）选项，则页面设置为 UTF-8 编码格式，内容也可以正常输出，如图 20.2 所示。



图 20.1　编码格式不匹配导致输出内容乱码

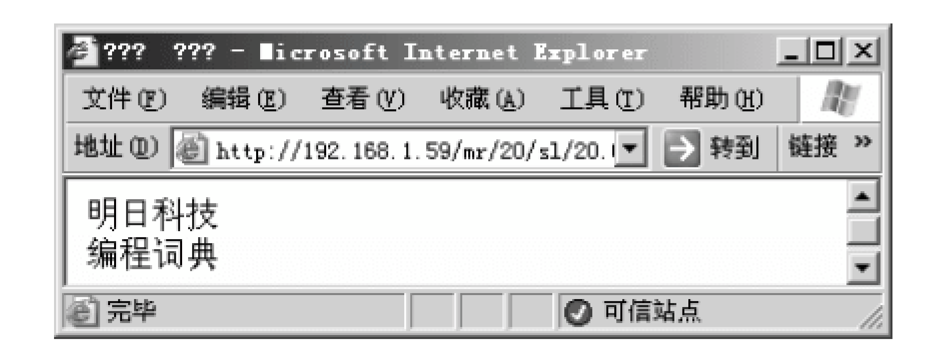


图 20.2　设置正确的编码格式后输出的内容

20.2.3　检测字符串的编码

在 PHP 中，可以通过 mbstring 扩展库中提供的 mb_detect_encoding()函数来检测字符串使用的是何种字符编码。

mb_detect_encoding()函数用于检测编码的格式。其语法如下：

```
mb_detect_encoding ( string str [, mixed encoding_list [, bool strict]] )
```

mb_detect_encoding()函数用于检测字符串 str 的编码格式，返回值为数据流的代码页（Code Page）。如果使用了参数 encoding_list，那么必须使用逗号进行分隔；如果省略了 encoding_list 参数，那么将使用默认的编码。

你问我答：什么是代码页（Code Page）？

代码页（Code Page）可以被理解为计算机的内码，GB2312 对应的 Code Page 是 EUC-CN，GBK 对应的 Code Page 是 CP936。

在操作系统中，使用代码页来满足不同国家和地区的文字使用需求。在简体中文的 Windows 下，默认保存文件的编码为 ANSI 或者默认编码，对应的代码页是 EUC-CN。如果保存的编码是 UTF-8，那么显示也是 UTF-8，所以可以得到来源数据的编码格式。

例如，通过 mb_detect_encoding()函数检测程序中指定字符串的编码格式，代码如下：

```
<?php
$str="获取源文件的编码格式";                       //定义变量
echo mb_detect_encoding($str)."\n";              //输出编码格式
?>
```



运行结果为：UTF-8

你问我答：浏览器是如何区分字符编码的优先级的？

由于字符编码定义的方法很多，所以当使用多种方式时就存在一个优先级的问题。下面给出浏览器在判断一个文本的解码方式时的优先级别（由高到低）。

（1）HTTP 消息报头的 Content-Type 字段中的 charset 参数。

（2）通过 meta 元素声明，将 http-equiv 设置成 Content-Type。

（3）一些元素的 charset 属性设置。

除了这些设置外，用户浏览器也可以自发地试探。例如，一些浏览器默认的字符编码方式是 ISO-8859-1，如果没有定义这些设置，浏览器就会试着使用 ISO-8859-1 来解码文本。

除此之外，用户也可以手动改变编码。一般来说，浏览器的菜单栏中都会列出不同的字符编码方式，用户可以从中选择一个作为试探，如果文本显示正确就证明文本采用了这种编码方式；如果文本显示错误，那么可以选择其他编码继续探索，直到文本显示正确。

20.3　PHP 开发中的乱码问题

PHP 开发中编码格式的设置并没有想象得那么复杂，只要保持文档的实际编码与编码指令一致，那么基本上就不会出现乱码的问题。下面对开发中出现乱码的问题进行分析。

20.3.1　解决页面中的乱码问题

如果在设计的页面中出现乱码，那么可以使用下面的两种方法来解决。

（1）使用 meta 标签设置页面编码。

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

meta 标签的作用是声明客户端的浏览器用什么字符集编码显示该页面，其中 charset 可以设置为 GB2312、GBK、UTF-8（和 MySQL 不同，MySQL 是 UTF8）等。大部分的页面都是通过这种方式来告诉浏览器显示这个页面时采用什么编码。

（2）通过 header()函数设置页面的编码格式。

```
header("content-type:text/html; charset=utf-8");
```

header()函数的作用是把括号内的信息发到 http 标头。通过 header()函数设置 charset 的值来指定页面的编码格式，其作用与 meta 标签是相同的。但不同之处是，如果同时使用 header()函数和 meta 标签来设置页面的编码格式，浏览器会采用 header()函数指定的编码格式，而不会使用 meta 标签设置的编码格式，这就是 header()函数的特殊之处。header()函数多数应用在纯 PHP 的文件中，通过它来设置文件的编码格式。

例 20.02　通过 header()函数和 meta 标签，分别为页面设置 UTF-8 编码和 GB2312 编码，输出一个中文字符串，看一下最终的输出结果是应用的哪种编码格式。具体步骤如下。**（实例位置：光盘\mr\20\sl\20.02）**

（1）通过 Dreamweaver 新建一个动态 PHP 文件，通过 header 区域中 meta 标签的 HTTP-EQUIV 属性，设置网页编码格式为 gb2312。

（2）编写 PHP 脚本，通过 header()函数设置页面编码格式为 UTF-8。

（3）编写 PHP 脚本，通过 echo 语句输出一个中文字符串。程序代码如下：

```
<?php
header("content-type:text/html;charset=UTF-8");                    //设置编码格式为 UTF-8
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>比较编码格式的设置</title>
</head>
<body>
<?php
echo "这天太冷了";
?>
</body>
</html>
```

运行结果如图 20.3 所示，由于页面中使用了 UTF-8 编码格式，所以导致输出内容为乱码。

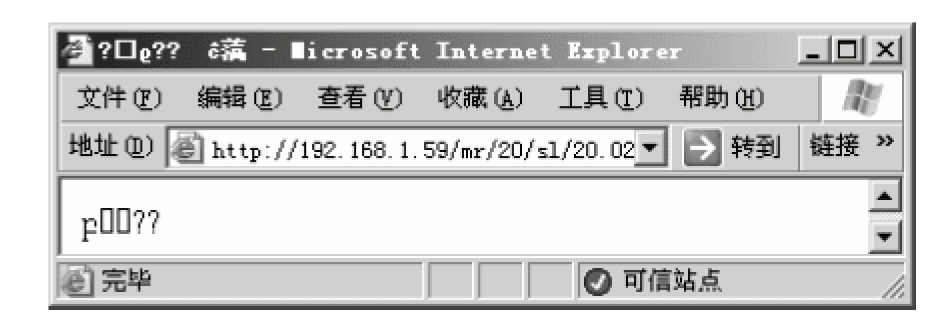


图 20.3　在 UTF-8 编码格式的页面中输出 GB2312 编码格式的字符串

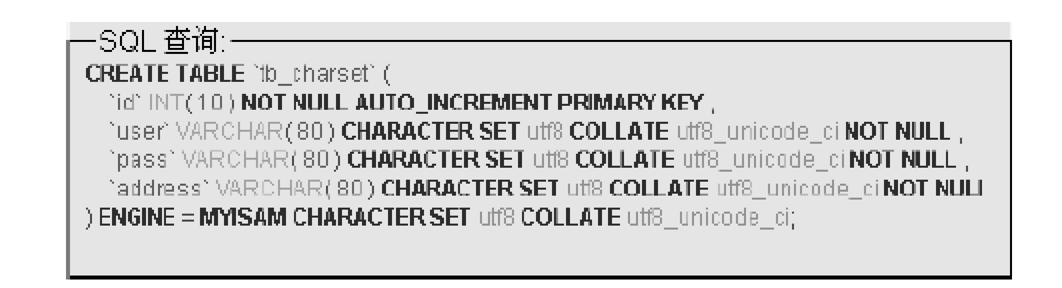
注意：header()函数只能在 PHP 脚本中使用，并且通过 header()函数设置页面为 UTF8 编码时，使用的格式是“UTF-8”。

20.3.2　数据库中的字符集编码问题

通过 PHP 操作数据库中的数据时，如果网页中使用的是 UTF-8 编码，那么就可以在连接数据库的文件中执行 mysql_query("SET NAMES UTF8")这个语句，指定数据库的编码为 UTF8，这样在读取出的中文数据就不会出现乱码的问题。也就是说，保证页面编码格式与数据库中输出数据的编码格式一致就不会出现乱码的问题。同样在向数据库中添加数据时，也要保证编码格式的统一。

例 20.03　下面编写一个实例，看一下在正确地使用编码格式和错误地使用编码格式输出数据库中的数据时会有什么不同。具体步骤如下。**（实例位置：光盘\mr\20\sl\20.03）**

（1）创建 db_database20 数据库，并创建一个包含 4 个字段的 tb_charset 数据表，设置数据库字符集为 UTF8。创建数据表的完整 SQL 语句如图 20.4 所示


SQL 查询:

```
CREATE TABLE `tb_charset` (
  `id` INT(10) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `user` VARCHAR(80) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,
  `pass` VARCHAR(80) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL ,
  `address` VARCHAR(80) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL
) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

图 20.4　创建数据表的 SQL 语句

（2）创建 conn.php 文件，完成与 db_database20 数据库的连接，并且通过 mysql_query()函数设置数据库的编码格式为 UTF8。代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or dir("服务器连接失败".mysql_error());    //连接数据库服务器
```



```
mysql_select_db("db_database20",$conn) or dir("数据库连接失败".mysql_error());          //连接 db_database20
mysql_query("set names utf8");                                                          //设置数据库编码格式
?>
```

注意：在通过 mysql_query()函数设置数据库的字符集时，如果要设置数据库的字符集为 UTF8，那么使用的格式是 “UTF8”。它与通过 header()函数设置字符集为 UTF8 是不同的。

（3）通过 Dreamweaver 新建一个动态 PHP 文件，通过 header 区域中 meta 标签的 HTTP-EQUIV 属性，设置网页编码格式为 UTF-8。载入数据库连接文件，完成 MySQL 数据库中数据的循环输出。关键代码如下：

```
<?php
    include_once("conn/conn.php");                              //载入数据库连接文件
    $query=mysql_query("select * from tb_charset",$conn);       //执行查询语句
    while($myrow=mysql_fetch_array($query)){                    //循环输出查询结果
?>
  <tr>
     <td align="center" bgcolor="#FFFFFF"><?php echo $myrow['id'];?></td>
    <td height="20" align="center" bgcolor="#FFFFFF"><?php echo $myrow['user'];?></td>
    <td bgcolor="#FFFFFF"><?php echo $myrow['pass'];?></td>
</tr>
<?php }?>
```

运行结果如图 20.5 所示。此时设置的数据库字符集与页面编码格式是相同的，所以内容可以正确地输出。

但是，如果在 conn.php 中将数据库字符集设置为 gb2312，那么再次运行本程序时，将输出如图 20.6 所示的内容，中文字符串将出现乱码。如果在 conn.php 中将设置数据库字符集的操作注释掉，那么在运行本程序时输出的中文字符串同样会出现乱码。

ID	用户名	密码
1	明日	123456
2	科技	456789

图 20.5　正确地读取数据库中的数据

ID	用户名	密码
1	÷□	123456
2	?□?	456789

图 20.6　编码格式不统一导致输出内容乱码

20.3.3　避免截取中文字符串时出现乱码

这里向读者推荐 mb_substr()函数，通过它来对中文字符串进行截取，就可以避免在截取中文字符串时出现乱码。

mb_substr()函数，对字符串进行截取操作，并且支持中文字符串的截取。其语法如下：

```
string mb_substr ( string str, int start [, int length [, string encoding]] )
```

mb_substr()函数的参数说明如表 20.1 所示。

表 20.1　mb_substr()函数的参数说明

参　　数	说　　明
str	必要参数，指定操作的字符串
start	必要参数，指定截取的开始位置。参数 start 的指定位置是从 0 开始计算的，即字符串中的第一个字符表示为 0
length	指定截取的字符串长度
encoding	设置字符串的编码格式

例 20.04 为了更好地理解 mb_substr()函数的应用，下面编写一个实例，应用 mb_substr()函数对字符串进行截取，并且输出截取后的字符串，程序代码如下。**（实例位置：光盘\mr\20\sl\20.04）**

首先，定义字符串变量$str，变量值为“PHP 自学手册”。然后，通过 substr()函数对字符串进行截取，从第一个字节开始，截取 6 个字节。最后，通过 mb_substr()函数对字符串进行截取，同样从第一个字节开始，截取 6 个字节。关键代码如下：

```
<?php
$str="PHP 自学手册";                          //定义字符串变量
$m_sub=mb_substr($str,0,6,"UTF-8");           //截取 6 个字节，并且设置返回字符串的编码格式为 UTF-8
echo $m_sub."<br>";                           //输出结果
?>
```

运行结果为：PHP 自学

通过 mb_substr()函数截取字符串时，一个英文字符串即为一个字节，而一个中文汉字同样也定义为一个字节，这样就避免了在截取中文字符串时出现乱码的问题。这也就是在本实例中文为什么通过 mb_substr()函数对字符串“PHP 自学手册”进行截取时返回“PHP 自学”这样一个值的原因。它将“PHP”定义为 3 个字节，空格为一个字节，“自学”为两个字节。

20.4 照猫画虎——基本功训练

20.4.1 基本功训练 1——Dreamweaver 中编码格式的选择

视频讲解：光盘\mr\20\lx\Dreamweaver 中编码格式的选择.exe

在通过 Dreamweaver 开发 PHP 项目时，首先应该确立默认创建文件的编码，完成编码的统一之后，就能够避免一些不必要的乱码问题出现，那么如何在 Dreamweaver 开发工具中设置默认的文件编码格式呢？

具体步骤如下：

（1）打开 Dreamweaver 开发工具，选择“编辑”/“首选参数”命令，将弹出如图 20.7 所示的页面。

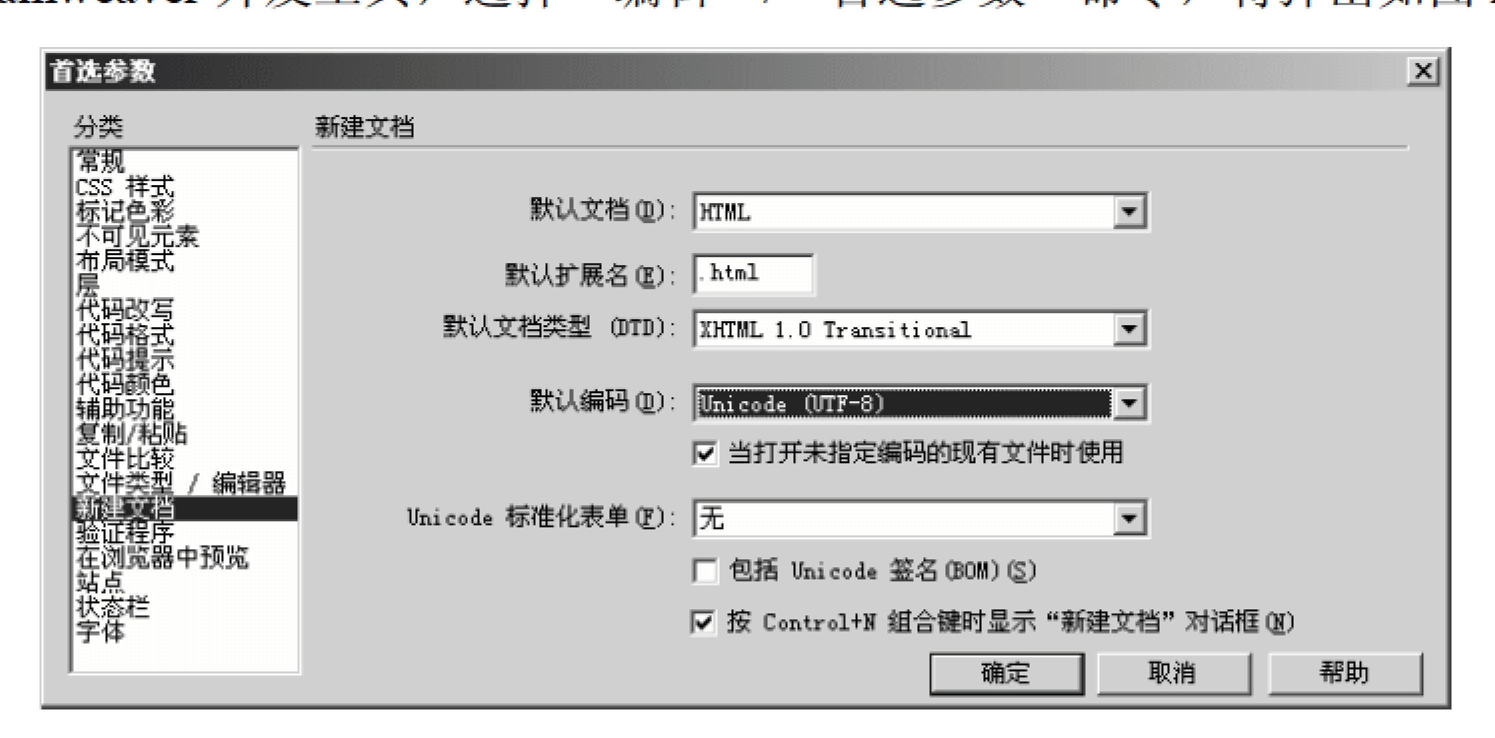
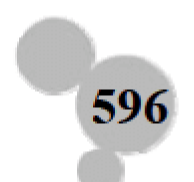


图 20.7 设置文件的编码格式

（2）在图 20.7 所示的页面中指定默认编码，最后单击“确定”按钮。

（3）在完成编码格式的设置之后，新建的文件将使用 utf-8 编码，如图 20.8 所示。



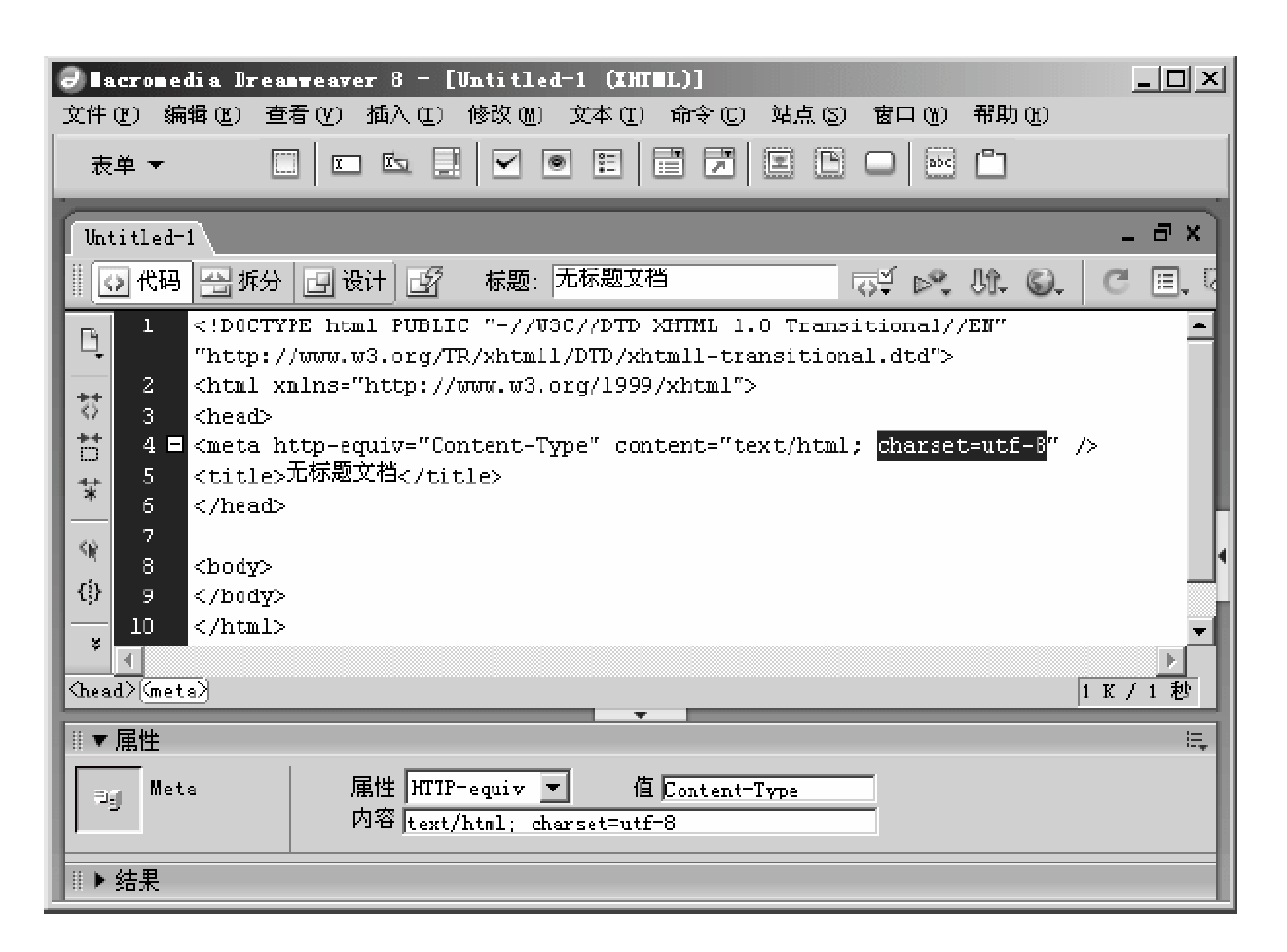


图 20.8　新建的网页文件

注意： 如果在默认编码格式为 utf-8 的情况下，通过 Dreamweaver 打开一个编码格式为 gb2312 的文件，其中输出的中文字符串将是乱码。此时需要将 Dreamweaver 默认的编码格式调整为 gb2312，然后再打开编码格式为 gb2312 的文件。

照猫画虎：根据上面的方法，设置 Dreamweaver 默认的编码格式为 gb2312。（20 分）

20.4.2　基本功训练 2——创建 gb2312 编码格式的网页

视频讲解：光盘\mr\20\lx\创建 gb2312 编码格式的网页.exe

在 Dreamweaver 中，既然要创建 gb2312 编码格式的网页，那么就必须先将 Dreamweaver 中默认的编码格式设置为 gb2312。然后，在 Dreamweaver 开发工具中，选择“文件”/“新建”命令，在弹出对话框的“类别”栏中选择“动态页”选项，在“动态页”栏中选择 PHP 选项，最后单击“创建”按钮，完成 gb2312 编码格式网页的创建，如图 20.9 所示。



图 20.9　创建 gb2312 编码格式网页

创建的 gb2312 编码格式的网页如图 20.10 所示。

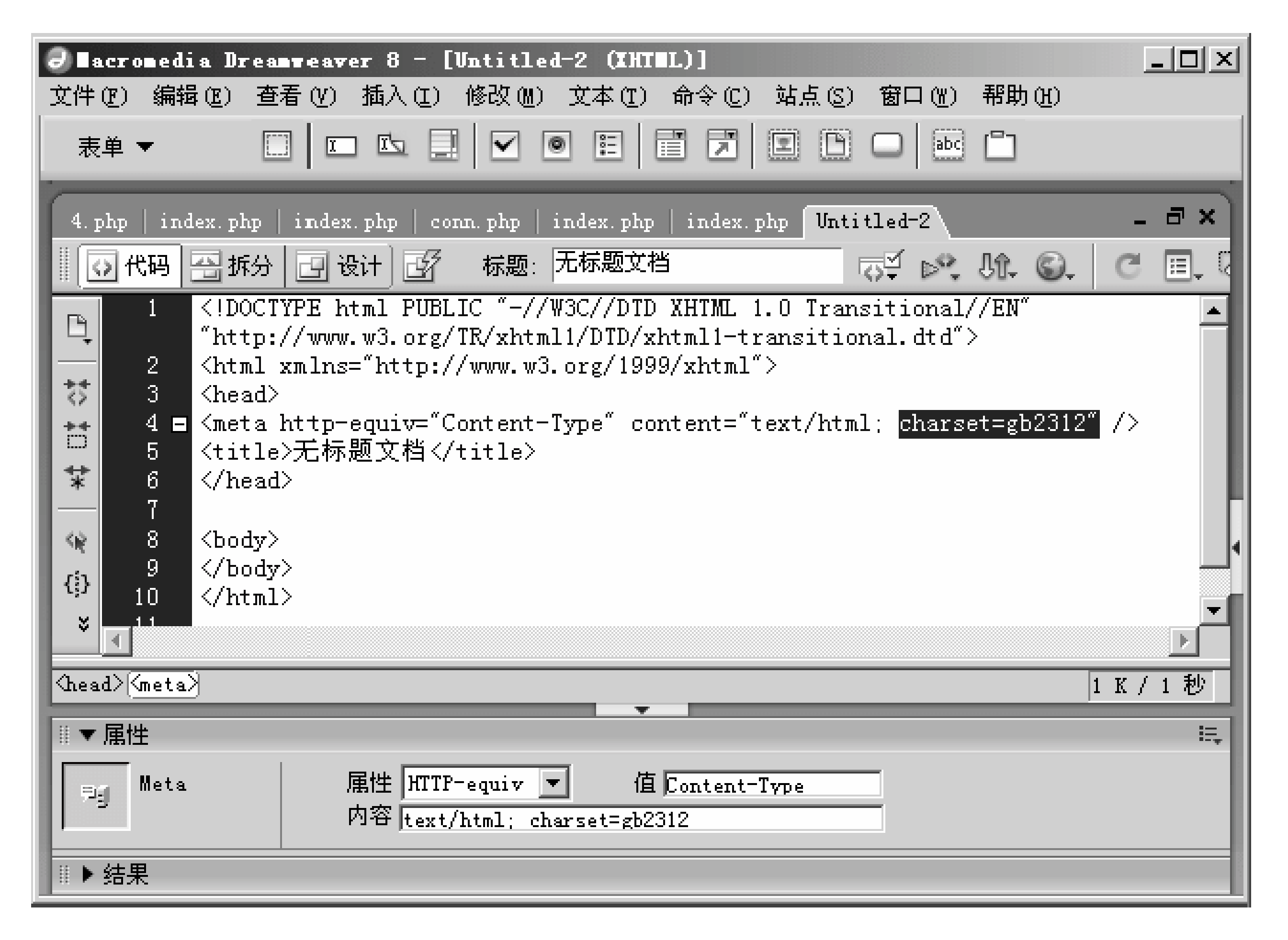


图 20.10　新建的 gb2312 编码格式的网页

照猫画虎：创建一个编码格式为 gb2312 的网页，在网页中编写一个 PHP 脚本，输出中文字符串“北京欢迎您！”。（20 分）

20.4.3　基本功训练 3——通过 iconv()函数实现编码格式的转换

视频讲解：光盘\mr\20\lx\通过 iconv()函数实现编码格式的转换.exe

实例位置：光盘\mr\20\zmhh\01

在 Dreamweaver 开发工具中，创建一个 utf-8 编码格式的网页，应用 PHP 中的 setlocale()函数完成区域化设置，输出美式的系统当前时间和中文的系统当前时间。在输出中文的系统当前时间时，因为 strftime()函数的返回值为 gb2312 编码，而当前页面使用的是 utf-8 编码格式，所以直接输出将会出现乱码，如图 20.11 所示。

所以必须应用编码格式转换函数对它的返回值进行编码格式转换才能够正常输出中文格式的时间，这里应用 iconv()函数对 strftime()函数的返回值进行编码格式转换，转换后的运行结果如图 20.12 所示。

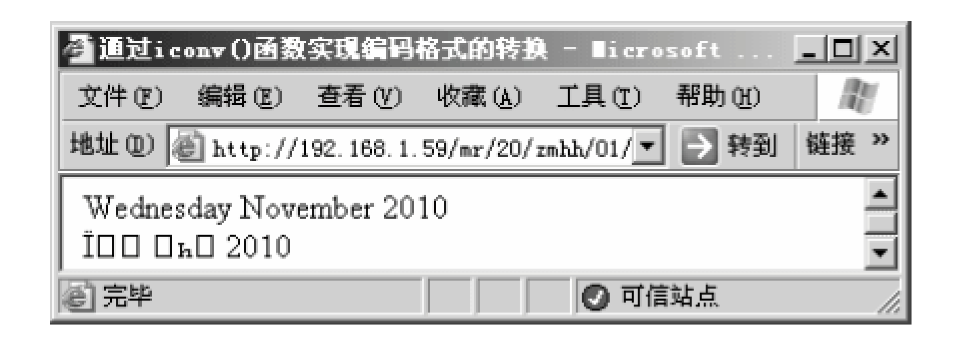


图 20.11　输出中文时间乱码



图 20.12　正常输出中文时间

本实例的代码如下：

```
<?php
setlocale(LC_ALL, 'en_US');              //设置为美式英语
echo strftime("%A %e %B %Y", time());    //输出当前时间
echo "<br>";
setlocale(LC_ALL, 'chs');                //设置为中文
```



```
$dates=strftime("%A %e %B %Y", time());//输出当前时间
echo iconv("gb2312","utf-8",$dates);        //输出经过编码格式转换后的中文时间
?>
```

照猫画虎：更改上述实例，采用 mb_convert_encoding()函数实现编码转换。（20 分）（实例位置：光盘\mr\20\zmhh\01_zmhh）

20.4.4　基本功训练 4——通过 header()函数设置 PHP 脚本文件的编码格式

视频讲解：光盘\mr\20\lx\通过 header()函数设置 PHP 脚本文件的编码格式.exe

实例位置：光盘\mr\20\zmhh\02

在 Dreamweaver 开发工具中，创建一个 utf-8 编码格式的网页，直接编写 PHP 脚本，通过 JavaScript 弹出一个提示对话框。如果没有设置这个 PHP 文件的编码格式，那么在弹出提示信息时，将输出如图 20.13 所示的效果。如果通过 header()函数设置 PHP 文件的编码格式为 utf-8，那么再次运行程序，将输出如图 20.14 所示的效果。



图 20.13　输出乱码



图 20.14　正常输出中文

本实例的代码如下：

```
<?php
header("Content-Type:text/html; charset=utf-8");      //设置文件的编码格式为 utf-8
echo "<script>alert('明日科技');</script>";            //弹出一个 JavaScript 对话框
?>
```

照猫画虎：更改上述实例，采用 mete 标签设置页面的编码格式，同样可以正常输出中文字符串。（20 分）（实例位置：光盘\mr\20\zmhh\02_zmhh）

20.4.5　基本功训练 5——设置 MySQL 数据库中数据的编码格式

视频讲解：光盘\mr\20\lx\设置 MySQL 数据库中数据的编码格式.exe

为了保证数据库中的数据与网页中编码格式的统一，在创建数据库和数据表时就完成对编码格式的统一，这样可以避免一些乱码的问题，不至于在向数据库中添加数据时出现乱码的问题。

具体步骤如下：

（1）在创建数据库时设置编码格式。这里以通过 phpMyAdmin 图形化管理工具为例进行讲解。当进入到 phpMyAdmin 图形化管理工具后，如图 20.15 所示。

（2）在图 20.15 中完成数据库的创建操作，并且设置数据库的编码格式，创建成功后将进入到图 20.16 所示的页面。

（3）在图 20.16 所示的页面中，添加数据表的名称，以及字段的数量，单击“执行”按钮，创建数据表，进入到图 20.17 所示的页面，完成数据表中字段的设置。

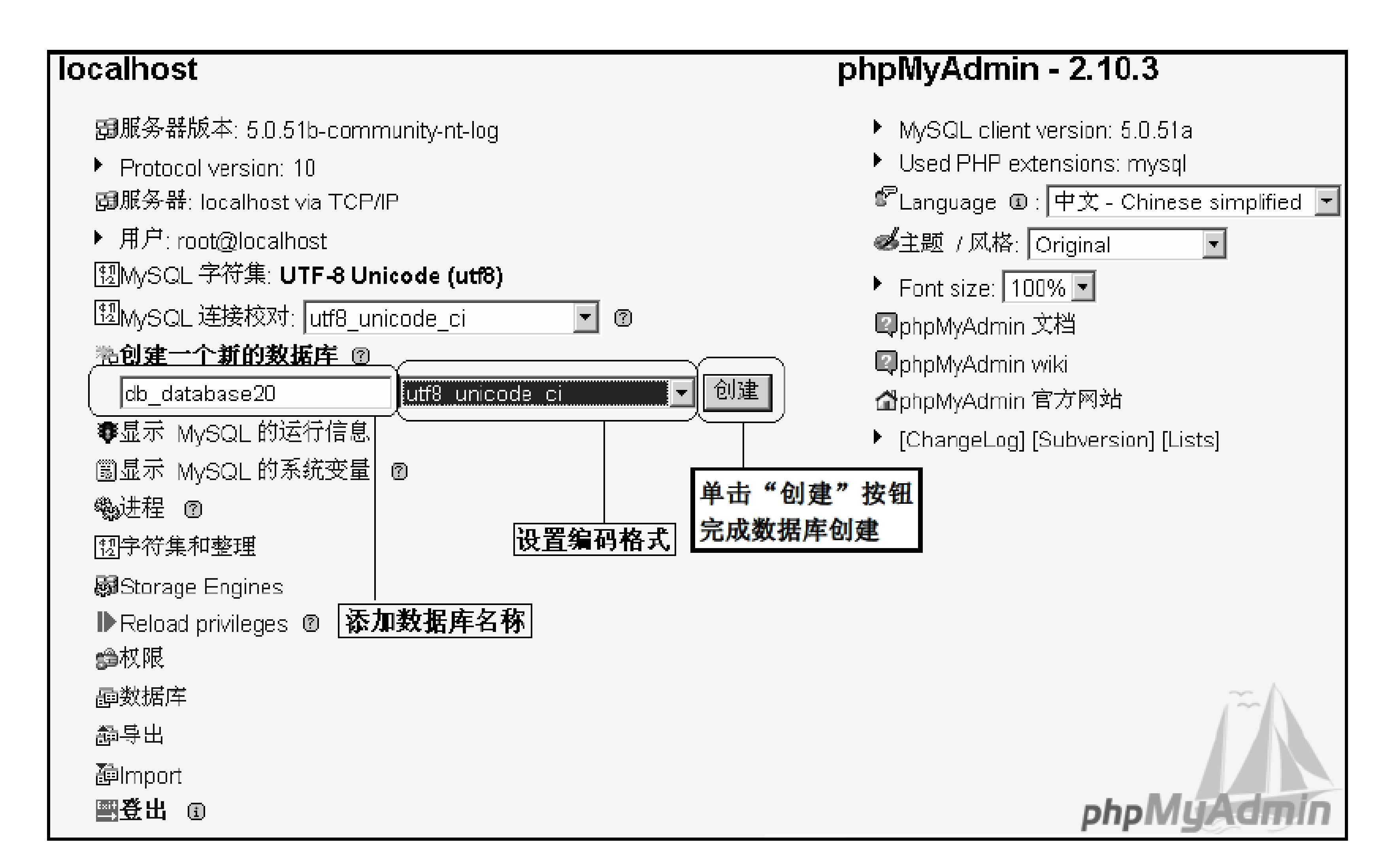
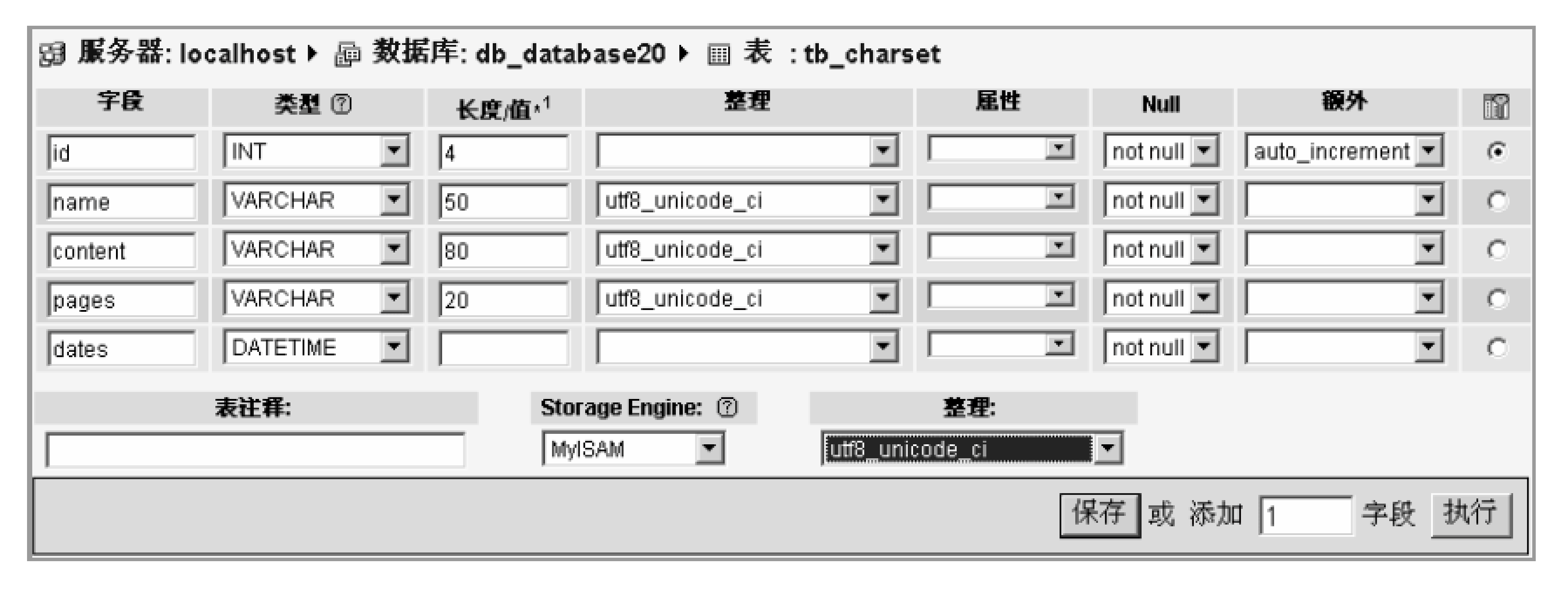


图 20.15　打开 phpMyAdmin 图形化管理工具



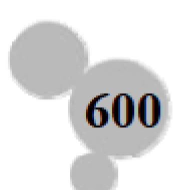
图 20.16　创建数据表



字段	类型	长度/值	整理	属性	Null	额外
id	INT	4			not null	auto_increment
name	VARCHAR	50	utf8_unicode_ci		not null	
content	VARCHAR	80	utf8_unicode_ci		not null	
pages	VARCHAR	20	utf8_unicode_ci		not null	
dates	DATETIME				not null	

图 20.17　设置数据表中的字段

（4）在设置数据表的字段时，可以对字符串类型的字段设置编码格式，同时最终指定数据表的编码格



式，确保数据库、数据表和字段的编码格式的统一。最后单击“保存”按钮，即可完成数据表的创建操作。

照猫画虎：通过本机的 phpMyAdmin 图形化管理工具，创建一个 gb2312 编码格式的数据库，并且设置数据表以及表中字段的编码格式都为 gb2312。（20 分）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

20.5　情景应用——拓展与实践

20.5.1　情景应用 1——采用 utf-8 编码设计论坛网页

视频讲解：光盘\mr\20\lx\采用 utf-8 编码设计论坛网页.exe

实例位置：光盘\mr\20\qjyy\01

设计一个论坛网页，采用 utf-8 编码格式，其运行结果如图 20.18 所示。



图 20.18　论坛页面效果图

设计一个 utf-8 编码格式的论坛页面的操作步骤如下：

（1）打开 Dreamweaver 开发工具，选择“编辑”/“首选参数”命令，在弹出的对话框中选择“新建文档”选项，查看新建文档的“默认编码”，如果不是 utf-8 编码，那么将其设置为 utf-8 编码，如果是则不需要修改，最后单击“确定”按钮。

（2）选择“文件”/“新建”命令，创建一个动态 PHP 页，其默认生成的代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>无标题文档</title>
</head>
<body>
</body>
</html>
```

（3）在默认生成的文件中完成论坛页面的设计，添加标题，通过表格完成页面的设计。关键代码如下：

```
<table id="__01" width="761" height="559" border="0" cellpadding="0" cellspacing="0">
	<tr>
		<td colspan="3">
			<img src="images/bg_01.jpg" width="761" height="104" alt=""></td>
	</tr>
	<tr>
		<td rowspan="2">
			<img src="images/bg_02.jpg" width="26" height="387" alt=""></td>
		<td>
			<img src="images/bg_03.jpg" width="504" height="42" alt=""></td>
		<td rowspan="2">
			<img src="images/bg_04.jpg" width="231" height="387" alt=""></td>
	</tr>
	<tr>
		<td>
			<img src="images/bg_05.jpg" width="504" height="345" alt=""></td>
	</tr>
	<tr>
		<td colspan="3">
			<img src="images/bg_06.jpg" width="761" height="68" alt=""></td>
	</tr>
</table>
```

（4）页面设计完成后，保存文件，将其存储于服务器指定的路径下即可。这里将其存储于本机的 E:\AppServ\www\MR\20\qjyy\01 文件夹下，并将其命名为 index.php。

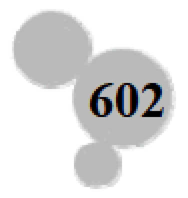
DIY：修改本机 Dreamweaver 的首选参数，创建一个 gb2312 编码格式的论坛页面。（20 分）（实例位置：光盘\mr\20\qjyy\01_diy）

20.5.2 情景应用 2——以 utf8 编码输出论坛中数据

视频讲解：光盘\mr\20\lx\以 utf8 编码输出论坛中数据.exe

实例位置：光盘\mr\20\qjyy\02

在通过 MySQL 函数库中的函数连接 db_database20 数据库时，使用 mysql_query()函数设置数据库的编码格式为 utf8，并且通过 while 语句和 MySQL 函数完成论坛中数据的循环输出，其运行结果如图 20.19 所示。





ID	标题	时间	积分
19	致明日科技	2009-11-16 09:40:40	100
15	PHP编程词典问世	2009-09-04 10:42:53	99
23	PHP编程词典问世	2010-08-19 03:12:22	99.6
24	PHP编程词典	2010-08-19 03:13:31	99.325

图 20.19　论坛页面效果图

具体步骤如下：

（1）在 Dreamweaver 中，设置新建文档的默认编码为 utf8，新建一个名称为 conn.php 的文件，完成与 MySQL 数据库的连接操作，并且设置数据库的编码格式为 utf8。关键代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or dir("服务器连接失败".mysql_error());     //连接数据库服务器
mysql_select_db("db_database20",$conn) or dir("数据库连接失败".mysql_error());          //连接 db_database20
mysql_query("set names utf8");                                                        //设置数据库编码格式
?>
```

（2）新建 index.php 文件，通过 include()语句包含数据库的连接文件，完成数据库中数据的循环输出，其关键代码如下：

```
<?php
     include_once("conn/conn.php");                                    //载入数据库连接文件
     $query=mysql_query("select * from tb_guestbook",$conn);           //执行查询语句
     while($myrow=mysql_fetch_array($query)){                          //循环输出查询结果
?>
      <tr>
            <td align="center" bgcolor="#FFFFFF"><?php echo $myrow['id'];?></td>
            <td align="center" bgcolor="#FFFFFF"><?php echo $myrow['title'];?></td>
            <td align="center" bgcolor="#FFFFFF"><?php echo $myrow['createtime'];?></td>
            <td align="center" bgcolor="#FFFFFF"><?php echo $myrow['integral'];?></td>
         </tr>
<?php }?>
```

DIY：修改本机 Dreamweaver 的首选参数为 gb2312，以 gb2312 编码输出论坛中的数据。提示：关键是在 conn.php 文件中，通过 mysql_query("set names gb2312")设置数据库的编码格式为 gb2312。（20 分）（实例位置：光盘\mr\20\qjyy\02_diy）

20.5.3　情景应用 3——论坛中控制帖子标题输出的长度

视频讲解：光盘\mr\20\lx\论坛中控制帖子标题输出的长度.exe

实例位置：光盘\mr\20\qjyy\03

在论坛中输出帖子的标题时，为了确保页面的美观、完整性，必须控制超长标题的输出，并以省略号

来替换，此时就需要对帖子的标题进行截取，这里推荐读者使用 mb_substr()函数对帖子的标题进行截取，可以避免截取中文时出现乱码的问题。本实例完成论坛中帖子标题的循环输出，并且对超长的标题进行截取，同时使用省略号进行替换，运行结果如图 20.20 所示。



图 20.20　论坛中控制帖子标题输出的长度

具体步骤如下：

（1）编写数据库连接文件 conn.php，设置数据库的编码格式为 utf8。

（2）新建 index.php 文件，通过 include()语句包含数据库的连接文件，完成数据库中数据的循环输出。通过 mb_strlen()函数统计帖子标题的长度，通过 if 语句判断如果帖子标题长度超过 5 个字节，那么通过 mb_substr()函数对帖子标题进行截取，并且通过省略号替换被截取的内容，否则正常输出帖子的标题。关键代码如下：

```
<?php
     include_once("conn/conn.php");                              //载入数据库连接文件
     $query=mysql_query("select * from tb_guestbook",$conn);     //执行查询语句
     while($myrow=mysql_fetch_array($query)){                    //循环输出查询结果
?>
      <tr>
            <td align="center" bgcolor="#FFFFFF"><?php echo $myrow['id'];?></td>
            <td align="center" bgcolor="#FFFFFF">
              <?php
              $str=mb_strlen($myrow['title'],"UTF-8");           //获取帖子标题的长度
              if($str>5){                                        //如果长度超出 5 个字节
                   $title=mb_substr($myrow['title'],0,5,"UTF-8"); //对帖子的标题进行截取
                   echo $title."...";                            //输出帖子标题
              }else{
                   echo $myrow['title'];                         //输出帖子标题
              }
              ?>
              </td>
            <td align="center" bgcolor="#FFFFFF"><?php echo $myrow['createtime'];?></td>
        </tr>
     <?php }?>
```

DIY：上述实例使用的是 utf-8 编码，在统计帖子标题的长度和进行截取时设置的编码格式为 utf-8。如果实例采用 gb2312 编码，那么要统计帖子标题的长度和进行截取时则设置编码格式为 gb2312。（20 分）（实例位置：光盘\mr\20\qjyy\03_diy）

20.5.4　情景应用 4——完成论坛中帖子的发布

视频讲解：光盘\mr\20\lx\完成论坛中帖子的发布.exe

实例位置：光盘\mr\20\qjyy\04

在论坛中帖子的发布功能是必须具备的，在将用户提交的帖子添加到数据库指定的数据表中时，必须注意帖子提交页的编码格式与数据处理页的编码格式是否统一，如果不统一很可能会导致帖子添加失败。本实例完成向论坛中添加帖子的操作，并且在数据的处理页中通过 header()函数设置页面的编码格式，从而保证数据的顺利提交。帖子提交成功的运行结果如图 20.21 所示。



图 20.21　帖子添加成功

具体步骤如下：

（1）编写数据库连接文件 conn.php，设置数据库的编码格式为 utf-8。

（2）创建 index.php 文件，设置页面的编码格式为 utf-8。添加表单，设置帖子标题和帖子内容的表单元素，将数据提交到 index_ok.php 文件。

（3）创建 index_ok.php 文件，通过 header()函数设置页面的编码格式为 utf-8。定义 insert 添加语句，将表单中提交的数据添加到指定的数据表中。关键代码如下：

```
<?php
header("Content-Type: text/html; charset=utf-8");             //设置页面的编码格式
include_once("conn/conn.php");
if(isset($_POST['title']) && isset($_POST['Submit'])){         //判断提交内容是否为空
    $sql="insert into tb_guestbook(title,content,createtime,integral)values('".$_POST['title']."','".$_POST['content']."',
'".date("Y-m-d H:i:s")."','0')";
    if(mysql_query($sql,$conn)){                                //执行添加操作
        echo "<script>alert('帖子添加成功！'); window.location.href='main.php';</script>";
    }else{
        echo "<script>alert('帖子添加失败！'); window.location.href='index.php';</script>";
    }
}else{
    echo "<script>alert('帖子内容不能为空！'); window.location.href='index.php';</script>";
}
?>
```

由于本实例中编写的文件应用的都是 utf-8 编码格式，所以在 index_ok.php 文件中必须通过 header()函数对文件的编码格式进行设置，如果不设置文件的编码格式，那么将导致程序运行出错，如图 20.22 所示。

出现此问题的原因是由于在 index_ok.php 文件中没有指定编码格式，导致中文字符串的输出乱码。所以在 index_ok.php 文件中必须指定其编码格式。

（4）创建 main.php 文件，循环输出数据库中最新的数据。



图 20.22 由于编码格式问题导致的错误

DIY：将上述实例完全修改为 gb2312 编码，那么在 index_ok.php 文件中就不需要使用 header()函数来设置页面的编码格式。提示：在连接数据库文件中需要将编码格式设置为 gb2312。（20 分）（实例位置：光盘\mr\20\qjyy\04_diy）

20.5.5 情景应用 5——对输出的数据进行编码格式转换

视频讲解：光盘\mr\20\lx\对输出的数据进行编码格式转换.exe

实例位置：光盘\mr\20\qjyy\05

PHP 不仅可以与 MySQL 数据库搭配使用，也可以与其他数据库进行联合应用。在本实例中，通过 PHP 操作 Access 数据库，完成数据库中数据的循环输出，并且通过 iconv()函数对输出数据的编码格式进行转换，运行结果如图 20.23 所示。

通过COM类连接ACCESS数据库

书名	出版社
《PHP从入门到精通》	清华大学出版社
《PHP项目开发全程实录》	清华大学出版社
《PHP开发实战宝典》	清华大学出版社

图 20.23 循环输出数据库中的数据

在本实例中，PHP 通过预先定义类 com 调用 ADO 方法操纵 Access 数据库。com 类的详细说明如下：

```
string com::com( string module_name [, string server_name [, int codepage]])
```

参数说明：

☑ module_name：被请求组件的名字或 class-id。

☑ server_name：DCOM 服务器的名字。

☑ codepage：指定用于将 PHP 字符串转换成 UNICODE 字符串的代码页，反之亦然。该参数的取值有 CP_ACP、CP_MACCP、CP_OEMCP、CP_SYMBOL、CP_THREAD_ACP、CP_UTF7 和 CP_UTF8。

具体步骤如下：

（1）建立数据库连接文件 conn.php，实现与 Access 数据库的连接，代码如下：



```
<?php
$conn = new com("adodb.connection");                    //实例化 com 类
$connstr="driver={microsoft access driver (*.mdb)}; dbq=". realpath("data/db_database20.mdb");
                                                        //连接 db_ database20 数据库
$conn->open($connstr);                                  //返回连接对象
?>
```

（2）创建 index.php 文件，包含数据库连接文件，调用 ADO 中的方法操作 Access 数据库中的数据，并通过 while 语句循环输出数据库中的数据。关键代码如下：

```
<?php
    $sql="select * from tb_book";              //定义 SQL 语句
    $rs=new com("adodb.recordset");            //实例化 ADO 中的方法
    $rs->open($sql,$conn,1,3);                 //执行 SQL 语句
    while(!$rs->eof){                          //循环输出查询结果
 ?>
  <tr>
    <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo iconv('gbk','utf-8',$rs->fields
('bookname')->value);?></div></td>
    <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo iconv('gbk','utf-8',$rs->fields
('pub')->value);?></div></td>
  </tr>
  <?php
     $rs->movenext;
   }
  ?>
```

技巧：由于本实例中的文件采用的都是 utf-8 编码格式，但是通过 ADO 中的方法读取到的 Access 数据库中的数据采用的是 GBK 的编码，所以在页面中直接输出时将出现乱码，因此必须通过 iconv() 函数对输出的数据进行编码格式的转换，才能保证数据的正常输出。

DIY：将上述实例完全修改为 gb2312 编码，那么在 index.php 文件中就不需要使用 iconv()函数对 ADO 方法中返回的数据进行编码格式的转换，可以直接输出。（20 分）（实例位置：光盘\mr\20\qjyy\05_diy）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

20.6　自 我 测 试

一、选择题（每题 10 分，5 道题）

1．Unicode 字符集是 Universal Multiple-Octet Coded Character Set 通用多（　　）编码字符集的简称。

　A．8 位　　　　B．16 位　　　　C．32 位

2．ISO 8859 是由 ISO（International Organization for Standardization，国际标准化组织）在（　　）编码

基础上制定的编码标准。

A．gb2312　　B．ASCII　　C．GBK　　D．UTF-8

3．下面不属于 UTF-8 优点的是（　　）。

A．128 以下编码和单字节处理软件兼容

B．UTF-8 的多字节编码没有部分字节混淆问题。例如，删除半汉字后整行乱码的问题在 UTF-8 中是不会出现的；任何一个字节的损坏都只影响对应的那个字符，其他字符都可以完整恢复

C．易于识别

4．在程序开发中转换编码集需要使用函数（　　）。

A．iconv　　B．string　　C．is_dir

5．下列编码可以被理解为计算机内码的是（　　）。

A．UTF-8　　B．GBK　　C．Code Page

二、填空题（每题 10 分，5 道题）

1．利用 meta 标记设置当前页面编码为 utf-8 的代码为（　　　）。

2．检测字符串编码格式需要使用函数（　　　）。

3．利用 header()函数设置页面的编码格式为 utf-8 的代码为（　　　）。

4．截取中文字符串时为解决乱码问题可以使用函数（　　　）来避免。

5．将字符串“PHP”的编码格式由“utf-8”转换为“gbk”的代码为（　　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

20.7 行动指南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	

续表

序号	内　　容	行 动 指 南
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）创建一个 gbk 编码格式的数据库和程序。
		（2）在博客中，控制标题输出的长度。
		（3）使用不同的方法对字符串进行截取操作。
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。	
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。	

20.8　成功可以复制——暴雪公司的领航者迈克·莫汉

成就：暴雪公司的创始人，世界十大游戏设计大师之一，魔兽争霸系列、星际争霸系列、魔兽世界系列游戏的开发者。

创始公司：暴雪公司（Blizzard Entertainment），全球最知名的电脑游戏软件公司。

财富巅峰：亿万富翁，世界最著名的游戏设计大师之一。

最高学历：大学

出生日期：1966 年

超级语录：优秀的游戏是我们的事业。

成功历程

暴雪公司 CEO 迈克·莫汉（Mike Morhaime）是一位相当具有传奇色彩的人物，他从只有 3 个办公桌的办公室起家，经过多年的打拼将暴雪发展成为一家全球知名的电脑游戏及电视游戏软件公司。他在暴雪的 20 年中，成功开发了魔兽争霸系列、星际争霸系列、暗黑破坏神系列以及《魔兽世界》游戏。魔兽争霸及星际争霸均被多项知名电子竞技比赛列为主要比赛项目。而单单《魔兽世界》游戏，每年就可以为暴雪公司带来高达将近 10 亿美元的收入。

迈克·莫汉在小学的时候就和哥哥妹妹们一起买了名为 Bally Professional Arcade 的游戏机，而他对于其上游戏的制作颇感兴趣。他在上学的时候就一直在研究如何让那些程序很好地运行。

作为美国加州大学电子工程系的学生，迈克·莫汉有更多的机会接触到游戏制作与开发，这也为他未

来创业奠定了基础。1991 年，大学毕业的迈克·莫汉（Mike Morhaime）和合作者共同创建了暴雪游戏开发公司。

一个大型游戏的开发，往往需要一年甚至更长的时间，投入与回报的周期比较长。初期，迈克·莫汉为了支付员工的薪水，经常需要从个人信用卡提取现金，依赖个人的借款来支撑公司。在那段时间里，他们的压力相当大，既要调动雇员的最大积极性发挥其才智，又要承受只出不入"耕耘期"的阵痛。当时，从资金流动的视角看，他们几乎一无所有。

1995 年对于迈克·莫汉来说是真正具有意义的一年，他们连续推出了《魔兽争霸 2》、《星际争霸》、《黑破坏神 2》等大作，受到了市场的追捧和热销。2007 年《魔兽世界：燃烧的远征》在发售后 24 小时内卖出了 240 万套，《魔兽世界》全球注册用户更是达到 1000 万。《魔兽世界》的成功使得迈克·莫汉成为享誉世界的游戏设计大师，也使他创办的暴雪公司一跃成为世界最顶级的游戏开发商，创造了游戏产业的一个神话。



《魔兽争霸》游戏画面

深度评价

借用 John Carmack 的话，更容易了解迈克·莫汉的成功：游戏的程序就是一行行的代码，需要你的灵魂才能赋予它以生命，有了生命的游戏才能带给大家快乐，也才能给你所想要的，你的游戏是你的热情，是你的汗水，是你的喜怒哀乐，是你的梦想；你的游戏，其实就是你自己。我从未度过没有编程的一天，这就是我的全部。

第21堂课

程序调试与错误处理

（视频讲解：112分钟）

在开发Web网站的过程中，经常会遇到很多问题，在遇到每个问题时都需要花很多的时间去解决。每当问题被解决后会发现其实这个问题根本没有当初遇到时那么棘手，只要有细心和耐心，问题就会很快地解决。

遇到问题解决问题，其实就是个人能力增长的过程。以一个圆圈为例，圆圈代表你的能力圈，中心点是你。当问题在能力圈内发生时，你都能迅速解决。如果产生了位于能力圈之外的问题，你不得不想办法去解决，但当你一一解决之后，能力圈也扩大了。这表示当你面临的问题愈多，能力就将愈强，因此遇到问题即是能力将有所突破的象征。无论你遇到任何问题，千万不要退缩。“问题”在拉丁文中的原意是“往前走”的意思。只要你往前走，问题的解决方法就会在不远处。

学习摘要：

- 基本调试策略
- 常见错误分析
- php.ini 文件中的错误处理机制
- 程序调试方法
- 错误处理技巧

21.1 基本调试策略

在使用 PHP 编写程序时，难免会出现一些错误，可能是复杂的逻辑错误，也可能是简单的语法错误。调试错误虽然不是什么高深的学问，但是有效的查找方式却可以缩短查找错误的时间。其基本的策略应该遵循以下原则，首先要判断错误最可能出现在哪一环节，然后针对该环节采取一些有效的措施来查找。基本调试流程如图 21.1 所示。

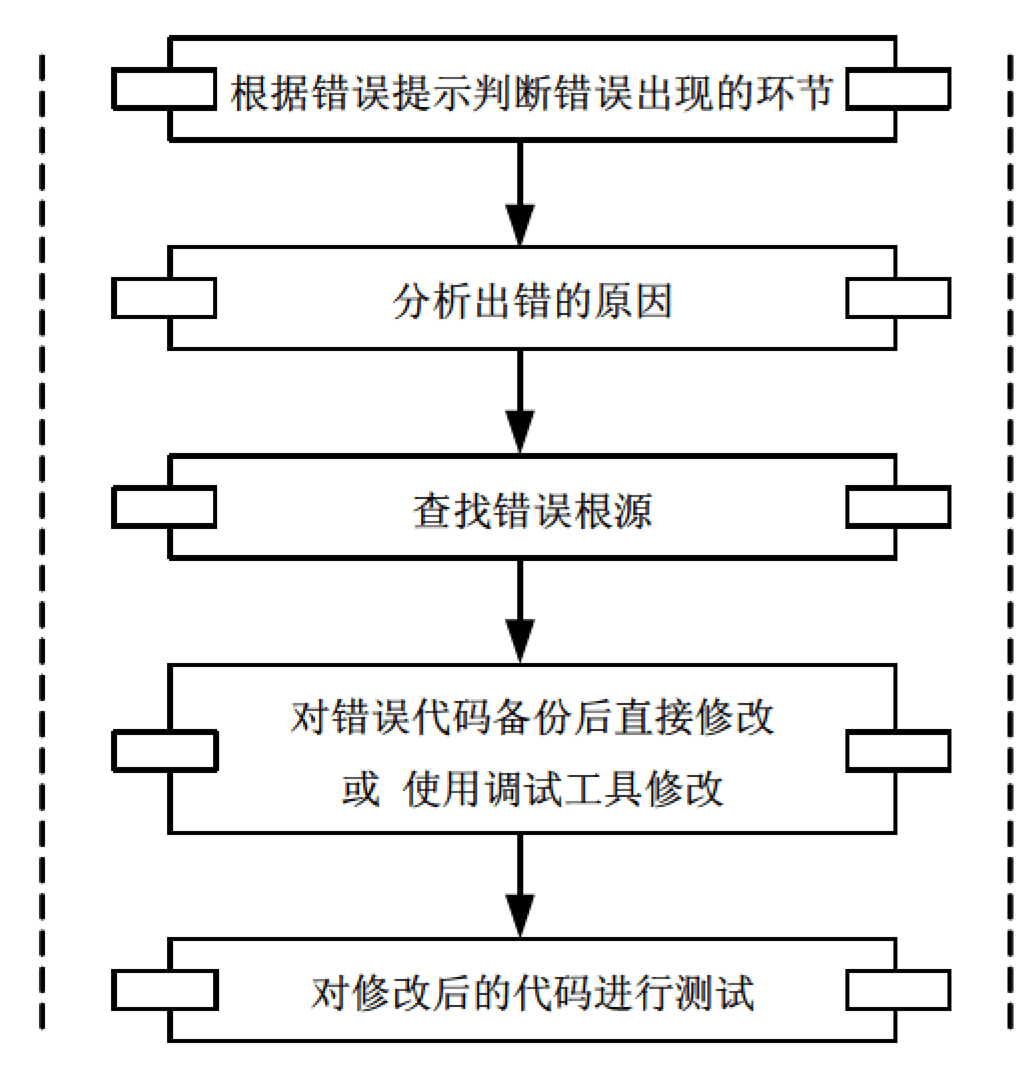


图 21.1　基本调试流程

上述这些只是进行错误调试工作的一个简单流程，具体的调试手段还是由程序设计者自己掌握，在这里介绍一些常用的调试手段。

（1）增加中间变量或跟踪变量。当程序结果与预期结果不一致时，可以通过增加中间变量，或者输出一些相关的变量值来发现错误根源。例如，在实现数据添加时，可以输出 SQL 语句来验证是否正确获取到数据值。

（2）应用注释语句排除法调试程序。当无法找到错误根源时，就可以应用注释符号（//或/* */）注释掉部分代码进行调试，即先注释掉部分代码，然后运行程序，查看错误是否存在，如果不存在，则说明错误在注释掉的那部分代码中；如果仍然存在，则继续对下一段代码进行注释，依此类推，直到查找到错误的代码为止。该方法是最常用的调试方法。

（3）通过调试器来单步调试，这样可以跟踪整个程序的执行过程，发现是否有些应该被执行的函数没有被执行，或者变量赋值错误等各种导致错误的原因。

21.2 常见错误分析

在程序的开发过程中，会产生很多的错误，尤其是初学者，对知识的掌握不够，以及经验的不足，出现错误是在所难免的。为了少走一些弯路，这里根据在学习过程中遇到的一些问题以及吸取别人的经验，



将 PHP 中常见的错误进行了总结，按照错误的类型可以分为语法错误、定义错误、逻辑错误、运行错误和环境错误 5 种。

21.2.1　语法错误

语法错误即 PHP 编译错误，在 PHP 编译过程中，一旦发生语法错误，程序就会立即终止执行。虽然语法错误出现的概率比较高，但是也非常容易解决，多数通过修正编写的代码就可以解决。下面介绍一些常见的语法错误。

1．缺少分号

在编写程序代码的过程中由于疏忽，在某个语句或者变量的结尾处没有使用结束符号“;”，从而导致程序出现语法错误。

例 21.01　创建两个变量，并且输出变量中的值，其中一个变量由于书写错误，缺少分号，代码如下。**（实例位置：光盘\mr\21\sl\21.01）**

```
<?php
$str="吉林省明日科技有限公司";            //正确的变量
$strs="mrsoft"                             //缺少分号
echo $str."<br>";
echo $strs;
?>
```

运行结果如图 21.2 所示。

2．缺少单引号或双引号

在编写程序代码的过程中，由于书写错误，使用单引号或者双引号时少书写一个，从而导致出现语法错误。

例 21.02　下面在程序中设置一个变量，并且输出一条语句，在语句的输出过程中少输入一个双引号，导致程序出错，程序代码如下。**（实例位置：光盘\mr\21\sl\21.02）**

```
<?php
$str="吉林省明日科技有限公司";            //正确的变量
$strs="mrsoft";
echo $str."<br>";
echo "明日科技;                            //缺少引号
?>
```

运行结果如图 21.3 所示。

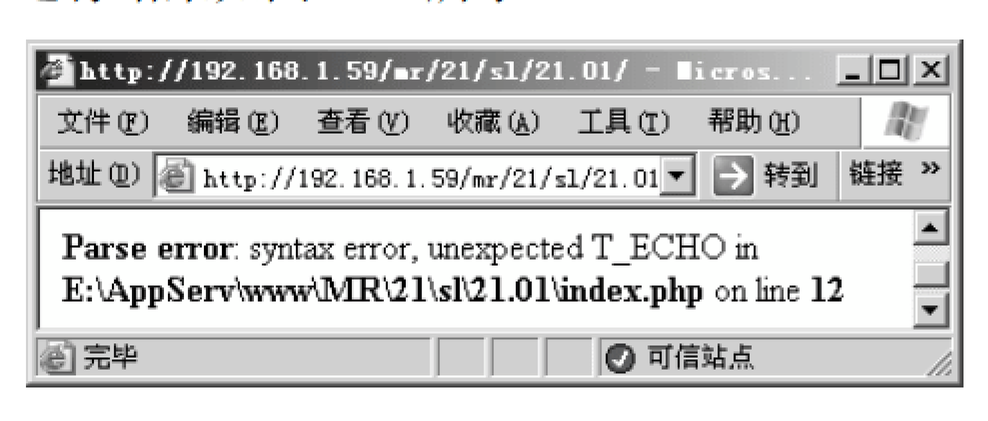


图 21.2　缺少分号

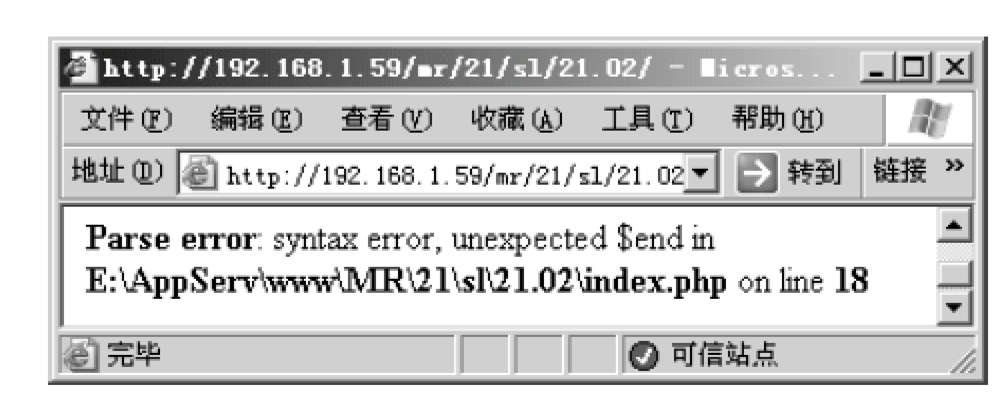


图 21.3　缺少引号

3．缺少关键字或者逻辑结构

在编写程序的过程中，一定要保证使用语句或者函数的书写完整性，否则也将导致语法错误。

例 21.03 下面应用 do…while 循环语句输出一个变量值，结果在使用 do…while 语句的过程中没有输出 while 关键字，从而导致程序语法错误，导致程序出错，程序代码如下。**（实例位置：光盘\mr\21\sl\21.03）**

```
<?php
$values=3;
do {   echo "变量的值是$values ";
        $values++ ;}
?>
```

运行结果如图 21.4 所示。

4．缺少括号

括号在程序中是经常使用的，也是经常导致错误的一个方面。由于括号的使用层数比较多，所以就可能导致括号的不匹配。

例 21.04 下面为实现一个多层括号嵌套应用的实例，看在括号不匹配时出现什么错误，程序代码如下。**（实例位置：光盘\mr\21\sl\21.04）**

```
<?php
$m=5 ;
$n=6 ;
$a=7 ;
$b=8 ;
if ((($m>$b) and ($m>$a)) or ($a>$b){        //其中缺少一个括号
        echo "括号使用正确!" ;
}else{
        echo "使用的不正确!";
}
?>
```

运行结果如图 21.5 所示。



图 21.4 缺少关键字



图 21.5 括号使用的不匹配

在大括号“{”后面有问题，但是其本身并没有问题，问题出在圆括号的使用不平衡上面，只需要修改成“if ((($m>$b) and ($m>$a)) or ($a>$b))”，然后保存就可以运行了。

5．缺少美元符号

在 PHP 中，设置变量时需要使用美元符号“$”，如果不添加美元符号就会引起解析错误。

例如，在 for 循环语句中，循环输出变量$i 小于 5 时的值，但是在定义条件表达式$i 时，忽略了美元符号“$”的输出。代码如下：

```
<?php
   for ( $i=0 ; $i<5; i++){        //缺少一个变量的美元符号
          echo "输出成功  $i!";
   }
?>
```

运行结果如图 21.6 所示。

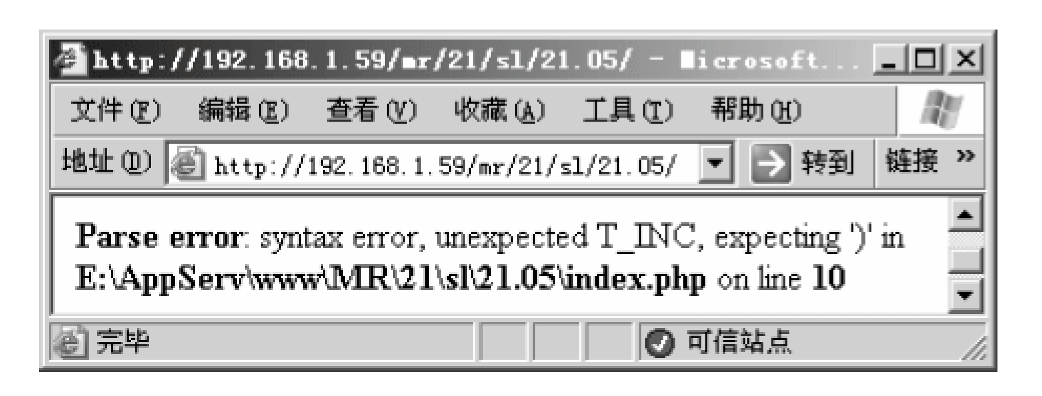


图 21.6　没有使用美元符号

在该语句中，“i++”应该修改为“$i++”，如果是前面的$i<5 没有使用美元符号，那么该程序将进入无限循环的状态，直到服务器终止执行。

技巧： 语法错误是最基本的错误，只要在编写程序时认真，那么就会减少很多麻烦。这里介绍的只是语法错误中的几种类型，还有很多类似这样的错误，要避免错误的出现就要在平时编写代码时注意，尽量书写完整准确的代码。在出现错误时，注意积累经验，避免同样的错误出现。

21.2.2　定义错误

定义错误是一种致命错误，引发此类错误的原因是由于调用一些不存在的类、变量或者函数等而导致程序终止运行。当出现此类错误时可以尝试在 PHP 或者 Apache 的 log 日志中查找错误的原因。

1．调用不存在的常量或者变量

在 PHP 中，如果调用一个没有声明的变量或者常量，将触发 NOTICE 错误。

例如，创建一个变量$str，并且通过 echo 语句输出这个变量的值，但是在输出时由于疏忽，将变量名称书写为$strs，查看其返回的错误信息。代码如下：

```
<?php
$str="明日科技";                    //定义变量
echo $strs;                         //输出时使用错误的变量名称
?>
```

运行结果如图 21.7 所示。

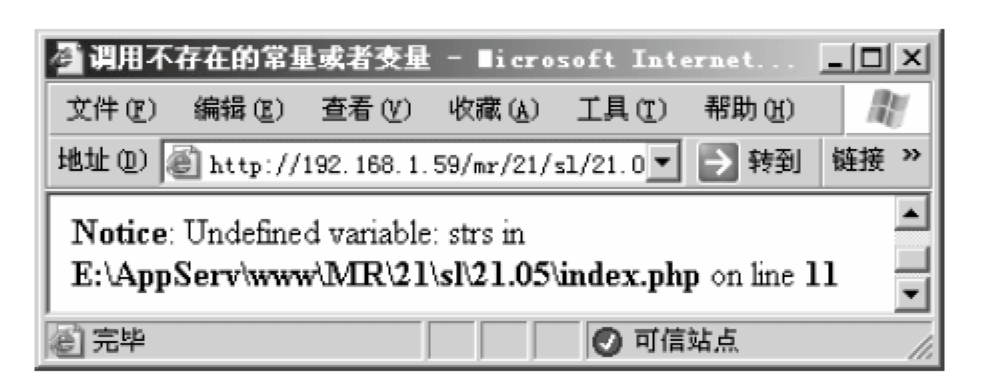


图 21.7　返回错误提示

如果程序运行后没有任何内容输出，原因是配置文件中错误级别的设置，在配置文件中设置显示程序中的致命错误，对警告信息不显示。

2．调用不存在的函数或者类

在 PHP 中，如果调用一个不存在的函数或者类，那么将终止程序的运行。

例 21.05　通过 mssql 函数库中的 mssql_connect 函数连接 SQL Server 数据库，但是 mssql 函数库没有被加载，将导致程序终止执行，并且返回错误信息。代码如下。**（实例位置：光盘\mr\21\sl\21.05）**

```
<?php
$conn = mssql_connect ( "localhost", "sa", "" ) or die ( "Connect SQL Server False" );    //连接 SQL Server 服务器
?>
```

运行结果如图 21.8 所示。

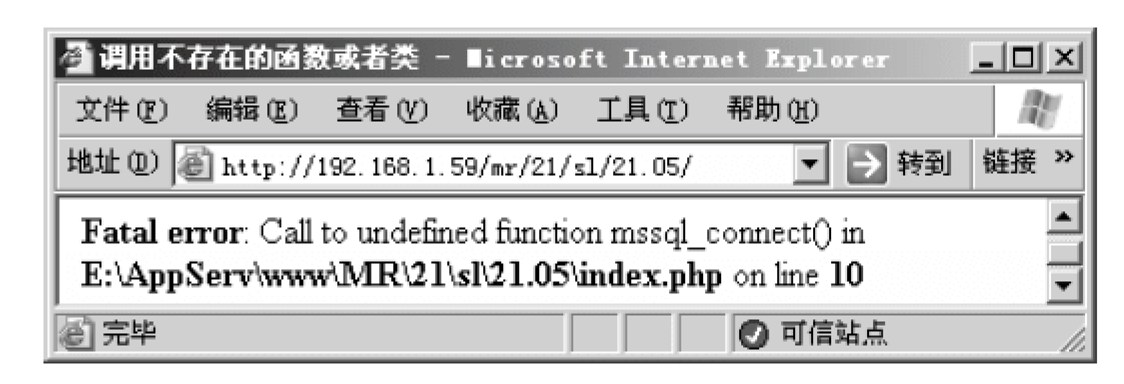


图 21.8　调用尚未被加载的函数

解决此类问题的方法是，在 php.ini 文件中加载所使用的类库（即去掉“;extension=php_mssql.dll”前面的分号），保存后重新启动 Apache 服务器即可。

21.2.3　逻辑错误

逻辑错误在语法上是不存在错误的，但是从程序的功能上看是 Bug。它是最难调试和发现的 Bug。唯一能看到的是程序的功能没有实现。对于逻辑错误而言，发现错误是容易的，但要查找出逻辑错误的原因却很困难。

例如，通过 print_r()函数输出数组中的第一个元素“php”。代码如下：

```
<?php
    $array=array("php","MySQL","apache");        //定义数组
    print_r($array[1]);                          //本意是输出数组中第一个元素
?>
```

但输出的结果却是“MySQL”，这就是一个逻辑错误，在定义位数组中元素的索引时，应该是从 0 开始，而不是从 1 开始。

21.2.4　运行错误

运行错误是指代码在运行时出现异常而导致的非致命错误，此类异常不是编程的逻辑错误造成的，而是 PHP 代码以外的问题，如磁盘、数据库、文件或者网络等。

运行错误是一件令人头痛的事情，很难检测到错误出现在什么地方，同时也更加难以修改。常见的运行错误举例如下。

1．操作只具备只读属性的文件

访问文件的错误也是经常出现的，例如，硬盘驱动器出错或写满，人为操作错误导致目录权限改变等。如果没有考虑到文件的权限问题，直接对文件进行操作就会产生错误。

例 21.06　下面看一个在读取文件时出现的错误，这里通过 fopen()函数打开文件 text.txt，并且向文件中写入“文件权限设置错误!”。但是由于 text.txt 文件权限设置为只读属性，所以在运行过程中出现错误，代码如下。**（实例位置：光盘\mr\21\sl\21.06）**

```
<?php
    $fp=fopen("text.txt","w");                   //以写入的格式打开文件
    fwrite($fp ,"文件权限设置错误!");             //写入数据
```

```
    fclose($fp);                                    //关闭文件
?>
```

运行结果如图 21.9 所示。



图 21.9　写入文件失败

由于该文件具有只读的权限，不能进行写入的操作。在执行这项操作时，首先要明确该文件的属性，是否是可写。如果要坚持执行操作，那就需要修改文件的权限。

2．包含的文件不存在

当使用 include()和 require()语句时，如果其中包含的文件不存在，就会产生错误。

例如，应用 include()语句调用一个不存在的文件 text.txt，代码如下：

```
<?php
    include ("text.txt") ;   //调用一个不存在的文件
?>
```

将输出如图 21.10 所示的错误提示。



图 21.10　包含的文件不存在

3．数据库服务器、数据库、数据表和数据引发的错误

在开发程序的过程中，对数据库的操作是非常多的，这也是出现错误频率比较高的地方。例如，在连接数据库服务器时，用户名或者密码设置不正确；连接不存在的数据库；在执行查询时数据库中没有数据，数据库文件只具备只读属性等，这些都可能导致程序在运行过程中出错。

例 21.07　通过 mysql_connect()函数连接 MySQL 数据库服务器，用户名是 root，密码是 111。但是在编写程序的过程中，将密码设置为 root。代码如下。（**实例位置：光盘\mr\21\sl\21.07**）

```
<?php
$conn=mysql_connect("localhost","root","root");                //连接 MySQL 服务器
?>
```

运行结果如图 21.11 所示。



图 21.11 连接 MySQL 服务器的密码设置不正确

21.2.5 环境错误

环境错误是与脚本的运行环境和相关服务关联的问题，如操作系统、PHP 版本、PHP 配置等。这种错误表现为数据库服务器不可用、文件无法打开、不具备操作权限等。

1. 操作系统

虽然 PHP 语言本身可以适应各种不同的操作平台，但是，这并不意味着在不同的操作系统下运行时没有问题出现。例如，在 PHP 函数库中，某些函数就是在特定的操作系统中使用的。另外，PHP 的功能在不同的操作系统下也略有不同；外部程序或者服务无法运行于所有平台；不同的操作系统采用的路径格式也不同，对文件名大小写的区分要求也不一致，这些都有可能导致程序在运行过程中出错。

2. PHP 版本

PHP 的版本经历了不断的更新之后，虽然高版本可以兼容低版本，但是如果将 PHP 5 的程序拿到低版本中运行，就很可能出现因版本不兼容而导致程序无法正常运行的问题。这是最令人头痛的一个问题，解决起来非常麻烦。要么更新服务器环境，要么对程序进行修改，无论使用何种方法，都有可能引发一些联动的问题。

3. PHP 配置

PHP 的配置非常灵活，给使用者提供了非常大的自由空间，但是这也带来了一些问题。由于每台计算机的配置方法各不相同，当程序更换运行环境之后就会因为配置文件的设置而无法正常运行。

你问我答：为什么要将 php.ini 文件中全局变量 register_globals 的值设置为 Off?

如果将 php.ini 文件中 register_globals 的值设置为 On，开启全局变量，那么在获取程序中表单提交的值、SESSION 变量传递的值以及超链接传递的参数值时，就可以直接使用变量名称来获取，即$username 等同于$_POST'username']、$_SESSION['username']、$_GET['username']。
但是当全局变量关闭（Off）之后，那么要获取表单提交的值、SESSION 变量传递的值以及超链接传递的参数值时，就必须使用$_POST['username']、$_SESSION['username']、$_GET['username']。
虽然开启全局变量可以为编写程序带来便利，但是同时也带来了安全隐患，所以应该尽量避免使用 register_globals。

你问我答：allow_url_fopen 控制文件系统函数是否可以操作远程文件？

对 allow_url_fopen 参数的设置决定是否允许打开 URL 标识的远程文件，在默认情况下被设置成开启状态 allow_url_fopen=On。此时程序就可以对远程的文件进行访问，但是当它的值设置为 Off 时，那么就不可以进行远程访问。

对于远程文件的访问操作，在服务器中多数情况下是不支持的，因为作为服务器它要考虑整个服务器系统的安全。所以，不建议读者应用远程文件的访问技术开发程序。

21.3　php.ini 中的错误处理机制

PHP 最基本的调试策略是使用 PHP 的错误报告机制，打印出简单的错误信息，并显示出错文件的行号。合理地运用 PHP 的错误处理机制可以降低程序调试的难度，提高开发效率，增强系统的稳定性。

注意：在程序的生成环境中，对于 PHP 的错误处理机制，还是关闭的好，不单是为避免浏览者看到一些错误信息，同时也避免给程序的安全带来隐患。

21.3.1　控制错误显示以及显示方式

在 php.ini 文件中可以控制错误是否显示，以及以何种方式显示。下面对这些设置方法进行详细介绍，包括配置选项的名称、默认值和表述的含义，如表 21.1 所示。

表 21.1　php.ini 文件中控制错误显示的配置选项说明

名　称	默 认 值	说　明
error_reporting	E_all &～E_notice	设定错误报告级别。其默认值表示除了通告以外的所有错误都可以显示。其参数是 PHP 错误级别中的任意常量，或者类似于默认值 E_ALL&～E_NOTICE 这样的一个逻辑表达式
display_errors	On	控制错误是否作为 PHP 的一部分输出。建议在开发环境中使用其默认值，但是在生产环境中还是要将其设置为 Off，因为它会暴露一些安全信息，给系统安全带来隐患
display_startup_errors	Off	控制是否显示 PHP 启动时的错误
track_errors	Off	设置是否在变量$php_errormsg 中保存最近一个警告或错误信息
html_errors	On	控制是否在错误信息中采用 HTML 格式。建议在生产环境中将其设置为 Off
error_log	null	设置记录错误日志的文件。默认值是 null，错误发送到 Web 服务器日志，否则就写入指定的文件。但是，如果这个值设置为 syslog，那么就使用操作系统日志记录
log_errors	Off	控制是否应该将错误发送到主机服务器的日志文件
log_errors_max_len	1024	控制在 log_errors 选项开启时，生成错误信息的最大长度。如果错误信息超过指定长度，将被截断。参数值设置为 0 则表示可以设置成任意长度
ignore_repeated_errors	Off	指定记录错误日志时是否忽略重复的错误信息
ig_repeated_source	Off	指定是否在忽略重复的错误信息时忽略重复的错误项
report_memleaks	On	指定是否报告内存泄露。该设置只在以调试方式编译的 PHP 中起作用，并且还必须在 error_reporting 配置选项的值中使用 E_WARNING
report_zend_debug	On	指定是否报告 Zend 引擎的 debug 信息

21.3.2　控制错误的级别

PHP 中的错误级别在向开发者展示错误严重性的同时，还可以使开发人员对错误进行准确的定位。错

误级别的控制通过 php.ini 文件中的 error_reporting 配置选项进行配置，其具体的错误级别描述如表 21.2 所示。

表 21.2　PHP 中的错误级别描述

值	错误常量	说　明
1	E_ERROR	致命的运行错误，PHP 无法从错误中恢复。脚本执行终止
2	E_WARNING	报告运行中的非致命错误，但不终止脚本运行
4	E_PARSE	由语言解析器产生的编译时的错误，表示无法执行这个脚本
8	E_NOTICE	报告通告、注意，表示所做的事情可能是错误的
16	E_CORE_ERROR	这些错误包括在对语言运行和进行初始化时 PHP 产生的错误，使脚本无法执行
32	E_CORE_WARNING	这是在对 PHP 的运行环境进行初始化时出现的问题
64	E_COMPILE_ERROR	当脚本中有错误，使 PHP 无法执行脚本时 PHP 产生的编译器错误
128	E_COMPILE_WARNING	当脚本中有不严重的错误，且不妨碍脚本的正常操作时 PHP 产生的编译器错误
256	E_USER_ERROR	这是用户产生的错误，可以由程序员通过 trigger_error 函数引发。脚本执行被终止
512	E_USER_WARNING	这是用户产生的警告，可以由程序员通过 trigger_error 函数引发，脚本执行不被终止
1024	E_USER_NOTICE	这是用户产生的注意，可以由程序员通过 trigger_error 函数引发，脚本不被终止，默认情况下不显示该信息
2048	E_STRICT	PHP 5.0 新增的，用于表示不符合保证最大互操作性和未来兼容性的最新建议的编码方式或结构

每个常量都表示一种错误类型，错误可以被报告也可以被忽略。例如，如果指定了错误报告的级别为 E_ERROR，则表示只有出现致命错误才会报告。其中的这些常量可以用二进制算法结合起来，产生不同的错误报告级别。

在 php.ini 文件中，默认设置输出除提示信息之外的所有错误。

```
Error_reporting = E_ALL & ～(E_BOTICE)
```

在上述语句中，符号“&”表示几个条件之间的并列，而“～”表示否定。

如果想忽略警告信息，可以设定如下语句。

```
Error_reporting = E_ALL & (E_NOITICE) & ～(E_WARNING)
```

21.4　程序调试方法

在对 PHP 中的程序进行调试时，除了可以使用在 21.1 节中介绍的基本调试策略之外，还可以使用 DIE 和 PRINT 语句进行调试，如果是 MySQL 语句中的错误，还可以使用 mysql_error 语句来获取错误信息。

21.4.1　应用 DIE 语句调试

应用 DIE 语句调试程序是一种不错的选择，不但可以查找出错误的位置，而且可以输出错误信息。使用 DIE 语句进行程序调试时，查询出错误后会终止程序的运行，并在浏览器上显示出错之前的信息和错误信息。该语句最常用的地方就是在 MySQL 数据库服务器的连接中，通过 DIE 语句明确地知道是否已经与数据库建立连接。

例 21.08　下面连接 MySQL 数据库服务器，指定数据库用户名为 root，密码是 111，数据库名为 db_database21。代码如下。（**实例位置：光盘\mr\21\sl\21.08**）



```
<?php
$conn=mysql_connect("localhost","root","111") or die("服务器连接失败：".mysql_error());        //连接服务器
echo "服务器连接成功!<br>";
mysql_select_db("db_database19",$conn) or die ("数据库连接失败：".mysql_error());           //连接数据库
echo "数据库连接成功!";
mysql_query("set names utf8");                    //设置数据库编码格式
?>
```

运行结果如图 21.12 所示。

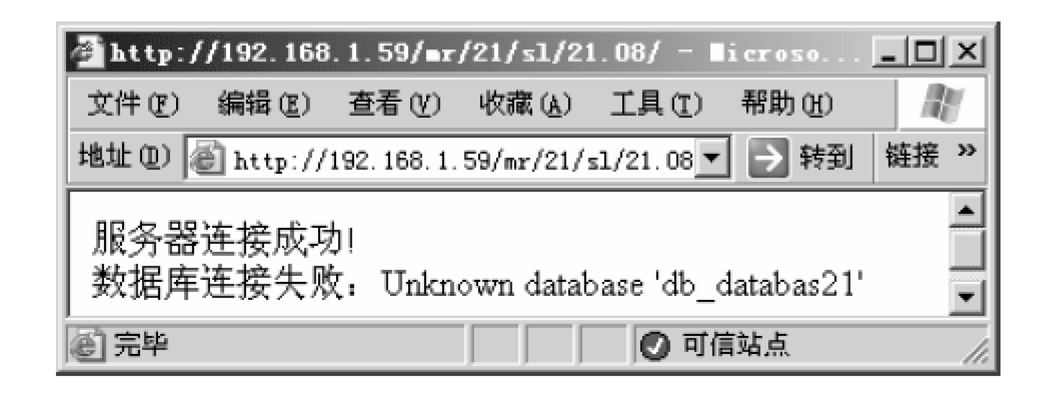


图 21.12　数据库连接失败

由于书写程序代码中的失误，将数据库名错写为 db_databas21，所以输出图 21.12 所示的结果。虽然服务器连接成功，但却没有找到指定的数据库。如果不使用 DIE 语句，则不会输出“数据库连接失败”的提示；如果不使用 mysql_error()函数，则不会输出“Unknown database 'db_databas21'”的提示。

21.4.2　应用 mysql_error()语句输出错误信息

在执行 MySQL 语句时产生的错误是很难发现的，因为在 PHP 脚本中执行一个 MySQL 的添加、查询、删除语句时，如果是 MySQL 语句本身的错误，程序中不会输出任何的信息，除非对 MySQL 语句的执行进行判断，成功输出什么，失败输出什么。

为了查找出 MySQL 语句执行中的错误，可以通过 mysql_error()语句对 SQL 语句进行判断，如果存在错误，则返回错误信息，否则没有输出，该语句的应用被放置于 mysql_query()函数之后。

例 21.09　下面通过 mysql_error()函数检测 SQL 语句中的错误。首先连接数据库，然后读取数据库中的数据，最后将数据库中的数据输出到浏览器中。代码如下。**（实例位置：光盘\mr\21\sl\21.09）**

```
<?php
        $sql="select * from tb_new";                    //定义查询语句
        $query=mysql_query($sql,$conn);                 //执行查询操作
        echo mysql_error();                             //获取 SQL 语句中的错误
        while($myrow=mysql_fetch_array($query)){        //循环输出查询结果
?>
```

运行结果如图 21.13 所示。



图 21.13　应用 mysql_error()语句输出 MySQL 语句中的错误信息

在本实例中由于书写上的失误，将数据表的名称“tb_news”写成“tb_new”，从而导致上述的错误。如

果在程序中不使用 mysql_error()函数，那么就不会输出“Table 'db_database21.tb_new' doesn't exist”。

21.5 错误处理技巧

在 PHP 的错误报告中会输出一些包含服务器信息的提示，在实际应用的环境中，由于一些环境原因导致的错误可能会给服务器或者 Web 系统带来安全隐患。因此，对于可能出现的错误的处理在实际应用环境中至关重要。

21.5.1 隐藏错误

PHP 提供一种隐藏错误的方法。即在被调用的函数名前加上“@”符号来隐藏可能由于这个函数导致的错误信息。

例如，在应用 fopen()函数打开文件时，如果出现文件不存在或者不可用等错误时，PHP 将会输出一条警告信息。关键代码如下：

```
<?php
$fp=fopen("foepn.text","r");                //以只读方式打开指定的文件
fclose();                                   //关闭指定的文件
?>
```

如果文件不存在，那么将输出如图 21.14 所示的结果。

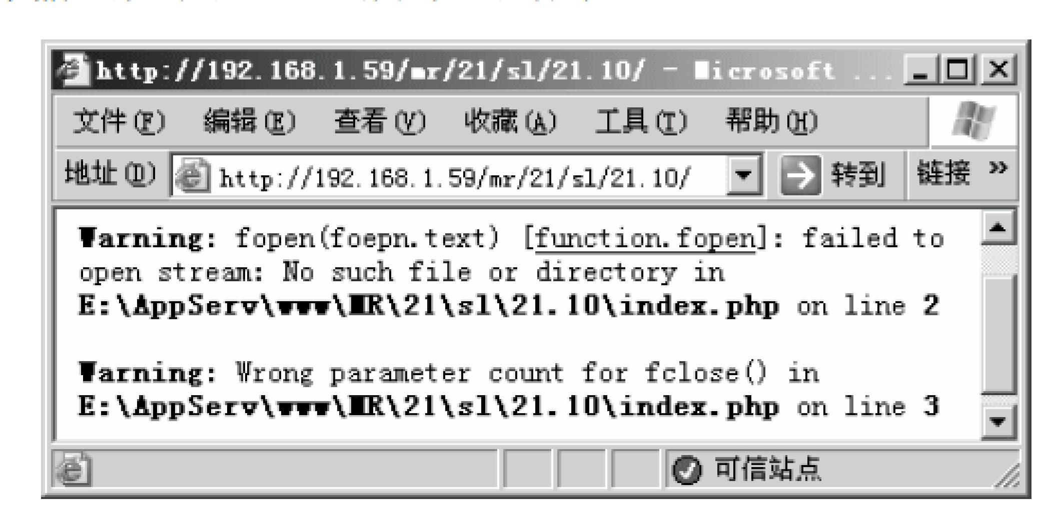


图 21.14　fopen()函数抛出的错误信息

从警告信息中可以看到，打开指定的文件失败，如果想要隐藏这个警告信息，就可以在 fopen()和 fclose()函数前加上“@”。修改后的代码如下：

```
<?php
$fp=@fopen("foepn.text","r");   //屏蔽错误信息
@fclose($fp);                   //屏蔽错误信息
?>
```

21.5.2 自定义错误页面

在 PHP 中，使用错误隐藏的方法来处理错误会另访问者很迷惑。因为访问者无法知道当前页面的状态，所以往往在隐藏错误信息的同时定制错误信息。

定制错误信息通常使用 if 语句来完成，判断当没有错误时执行什么内容，当出现错误时执行什么内容。

例 21.10　下面应用 if 语句定制错误，判断执行的文件是否被打开，如果打开失败则跳转到错误提示页面，代码如下。**（实例位置：光盘\mr\21\sl\21.10）**



```
<?php
if($fp=@fopen("foepn.txt","r")){            //判断文件打开的操作是否执行成功
    echo "文件打开成功！";                  //执行成功输出的内容
}else{
    header("Location:error.html");          //重定向页面
}
fclose($fp);                                //关闭文件
?>
```

当文件打开失败时，则跳转到如图 21.15 所示的错误页面。



图 21.15　自定义错误提示页面

自定义错误页面是在实际的生产环境中经常使用的一种提示错误信息的方法，该方法只给出一个错误的提示，并不具体给出是哪里出现了错误，从而可以避免访问者通过错误信息获取到程序中的一些重要信息。

21.6　照猫画虎——基本功训练

21.6.1　基本功训练 1——解决程序的语法错误

视频讲解：光盘\mr\21\lx\解决程序的语法错误.exe

实例位置：光盘\mr\21\zmhh\01

在应用 if、for、while 和 switch 等流程控制语句时，必须注意其中大括号的书写，它们必须是成对出现的，否则将导致程序运行出错，如图 21.16 所示。

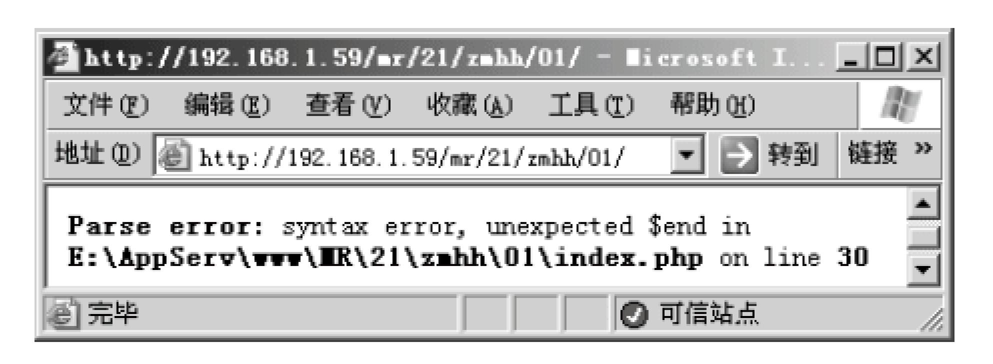


图 21.16　由于大括号不完整导致的错误

在本实例中，实现一个简单的网页计数器的功能，并且将数据存储到指定的文本文件中，关键代码如下：

```
<?php
if(($fp=fopen("counter.txt","r"))==false){
    echo "打开文件失败!";
}else{
    $counter=fgets($fp,1024);               //读取文件中的数据
    fclose($fp);                            //关闭文本文件
    $counter++;                             //计数器增加 1
    $fp=fopen("counter.txt","w");           //以写的方式打开文本文件<!---->
```

```
        fputs($fp,$counter);                        //将新的统计数据增加 1
        fclose($fp);
if(($fp=fopen("counter.txt","r"))==false){          //打开文件
        echo "打开文件失败!";
}else{
        $counter=fgets($fp,1024);                   //读取文件中的数据
        fclose($fp);                                //关闭文件
echo $counter;
}
?>
```

但是，在编写 if 条件语句时，少写了一个结束的大括号，结果导致程序运行出错。

要避免此类错误的发生，就要培养正确的书写习惯，在编写各种流程控制语句时，务必要保证它的完整性。例如，编写 while 语句（while(){}），if 条件语句（if(){}else{}）。如果每次都编写完整的语句内容，就可以避免出现缺少结束符号的问题。

照猫画虎：完善上面的网页计数器功能，添加缺少的结束大括号，使程序可以正常运行。（20 分）（实例位置：光盘\mr\21\zmhh\01_zmhh）

21.6.2　基本功训练 2——计算商品的折扣价格

视频讲解：光盘\mr\21\lx\计算商品的折扣价格.exe

实例位置：光盘\mr\21\zmhh\02

某商城实现商品优惠活动，如果用户为普通用户，那么商品不打折；如果是商城的会员，那么商品打八五折。这里实现的就是商品的折扣价格，代码如下：

```
<?php
if($user="普通用户"){
        echo $price=485*1;                          //485 是商品价格，1 是指不打折
}
echo "<br>";
if($user="会员"){
        echo $price=485*8.5;                        //485 是商品价格，8.5 是打八五折
}
?>
```

上面的代码对于 PHP 编译器来说没有任何问题。运行程序时，程序没有弹出错误信息。但是当用户为商城的会员时，商品价格乘以一个 8.5，这一点就没有符合要求，就属于逻辑错误，应该乘以 0.85 才正确。

对于逻辑错误而言，发现错误是容易的，但要查找出逻辑错误的原因却很困难。因此，在编写程序的过程中，一定要掌握所使用语句或者函数的功能、属性，避免出现逻辑错误。

照猫画虎：完善上面的商品折扣价格计算功能，正确地计算出会员的优惠价格。（20 分）（实例位置：光盘\mr\21\zmhh\02_zmhh）

21.6.3　基本功训练 3——连接不存在的数据库

视频讲解：光盘\mr\21\lx\连接不存在的数据库.exe

实例位置：光盘\mr\21\zmhh\03

在运行光盘中涉及数据库的程序时，经常会犯这样的错误，就是没有进行数据库的附加，而直接运行



程序，则导致输出如图 21.17 所示的错误信息。



图 21.17　没有找到指定的数据库

连接数据库的代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or die("数据库服务器连接失败：".mysql_error());
mysql_select_db("db_database21",$conn) or die("数据库连接失败：".mysql_error()) ;
?>
```

一旦出现上述的问题，首先要查看 MySQL 安装文件夹下的 data 目录，查看在这个目录下是否存在你所连接的数据库，如果不存在，就要附加或者创建这个数据库。还有一种情况，就是在编写数据库的连接代码时，书写的数据库名称与数据库中使用的名称不符，同样导致找不到指定的数据库。

照猫画虎：完成 db_database21 数据库的附加，同时去掉数据库文件的只读属性，然后向指定的数据表中添加数据。如果没有去掉数据库文件的只读属性，那么会导致数据添加失败。（20 分）（实例位置：光盘\mr\21\zmhh\03_zmhh）

21.6.4　基本功训练 4——运行缺少第三方组件的程序

视频讲解：光盘\mr\21\lx\运行缺少第三方组件的程序.exe

实例位置：光盘\mr\21\zmhh\04

在运行的光盘中涉及第三方组件的程序时，由于没有看光盘中提供的使用说明书，而直接运行程序，那么将导致程序运行出错，如图 21.18 所示。



图 21.18　没有找到指定的包含文件

在这个实例中，通过 Jpgraph 类库生成折线图，分析 2010 年公司销售额，并且在折线图中通过插入的图像设置数据点标签。但是在直接运行光盘中的程序时，却出现了如图 21.18 所示的错误，其原因就是没有认真地阅读程序使用说明书，因为本程序应用到一个第三方的组件 Jpgraph 类库，需要自行下载，并且将下载的 src 文件夹复制到实例的根目录下，然后才可以运行程序。

照猫画虎：下载 Jpgraph 类库，并且将其中的 src 文件夹复制到实例的根目录下，完成实例的正确运行。（20 分）（实例位置：光盘\mr\21\zmhh\04_zmhh）

21.6.5 基本功训练 5——通过 readfile()函数访问远程文件

视频讲解：光盘\mr\21\lx\通过 readfile()函数访问远程文件.exe

实例位置：光盘\mr\21\zmhh\05

在本实例中通过文件系统函数库中的 readfile()函数访问一个远程文件，但是由于本机 php.ini 文件的配置，不允许访问远程文件，结果导致程序运行失败，输出如图 21.19 所示的错误信息。

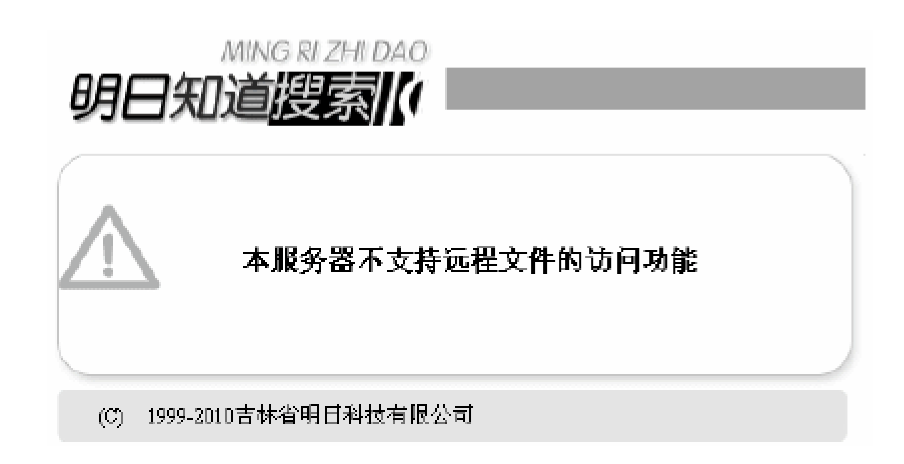


图 21.19 服务器不支持远程文件的访问

本实例的代码如下：

```
<?php
if(@readfile('http://192.168.1.235/','r')){    //判断文件打开的操作是否执行成功
}else{
    echo "<script>alert('您没有访问远程文件的权限！');window.location.href='error.html'</script>"; //重定向页面
}
?>
```

如果想让服务器支持远程文件的访问，必须修改 php.ini 文件，定位到 allow_url_fopen = Off 选项，将 allow_url_fopen 的值修改为 On，保存后重新启动 Apache 服务器即可。

照猫画虎：开启本机服务器远程文件的访问权限，通过 fopen()函数访问指定的远程文件，并且通过 fread()函数读取远程文件中指定的内容。（20 分）（实例位置：光盘\mr\21\zmhh\05_zmhh）

照猫画虎栏目分数统计：

照猫画虎题目	1	2	3	4	5	总分数	
分数							

21.7 情景应用——拓展与实践

21.7.1 情景应用 1——分析、解决 PHP 与 MySQL 连接错误

视频讲解：光盘\mr\21\lx\分析、解决 PHP 与 MySQL 连接错误.exe

实例位置：光盘\mr\21\qjyy\01

针对 MySQL 语句中出现的错误，这里对其进行分析，错误主要出现在以下 3 个环节中。

☑ 提交表单的创建。

☑ 对提交的数据处理。

☑ 存储到数据库。



在本实例中，首先创建 form 表单，通过 POST 方法将表单中的数据提交到 index_ok.php 文件中。在该文件中，完成与数据库服务器、数据库的连接，并且通过$_POST[]方法将表单中提交的数据添加到指定的数据表中。下面就通过这个实例对 PHP 操作 MySQL 数据库的过程中常见的错误进行分析。

第一个环节是表单提交页和数据处理页之间。在表单提交页中创建的表单的文本域名称一定要与提交到数据处理页的变量相一致，否则该文本域的值是不会被存储到数据库中的。对应关系如图 21.20 所示。

```
<form id="form1" name="form1" method="post" action="index_ok.php">
  <p>
    错误名称：
      <input name="log_name" type="text" id="log_name" size="25" />
  </p>
  <p>错误描述：
    <textarea name="log_content" cols="25" rows="3" id="log_content"></textarea>
  </p>
  <p align="center">
    <input type="submit" name="Submit" value="提交" />
      <input type="reset" name="Submit2" value="重置" />
  </p>
</form>
```

定义表单元素名称

定义表单元素名称

```
<?php
$conn=mysql_connect("localhost","root","111") or die("服务器连接失败：".mysql_error());
mysql_select_db("db_database19",$conn);       //连接数据库
mysql_query("set names utf8");                //设置数据库编码格式
if($_POST['Submit']==true && $_POST['log_name']!="" && $_POST['log_content']!=""){
$sql="insert into tb_log(log_name,log_content,log_date)
    values('".$_POST['log_name']."','".$_POST['log_content']."','".date("Y-m-d H:i:s")."')";
    $result=mysql_query($sql,$conn);                    //执行SQL语句
    echo mysql_error();                                 //获取SQL的错误信息
    if($result){                                        //判断添加操作是否成功执行
        echo "<script>alert('数据添加成功！');window.location.href='index.php';</script>";
    }else{
        echo "<script>alert('数据添加失败！');window.location.href='index.php';</script>";
    }
}else{
    echo "<script>alert('提交数据不能为空！');window.location.href='index.php';</script>";
}
?>
```

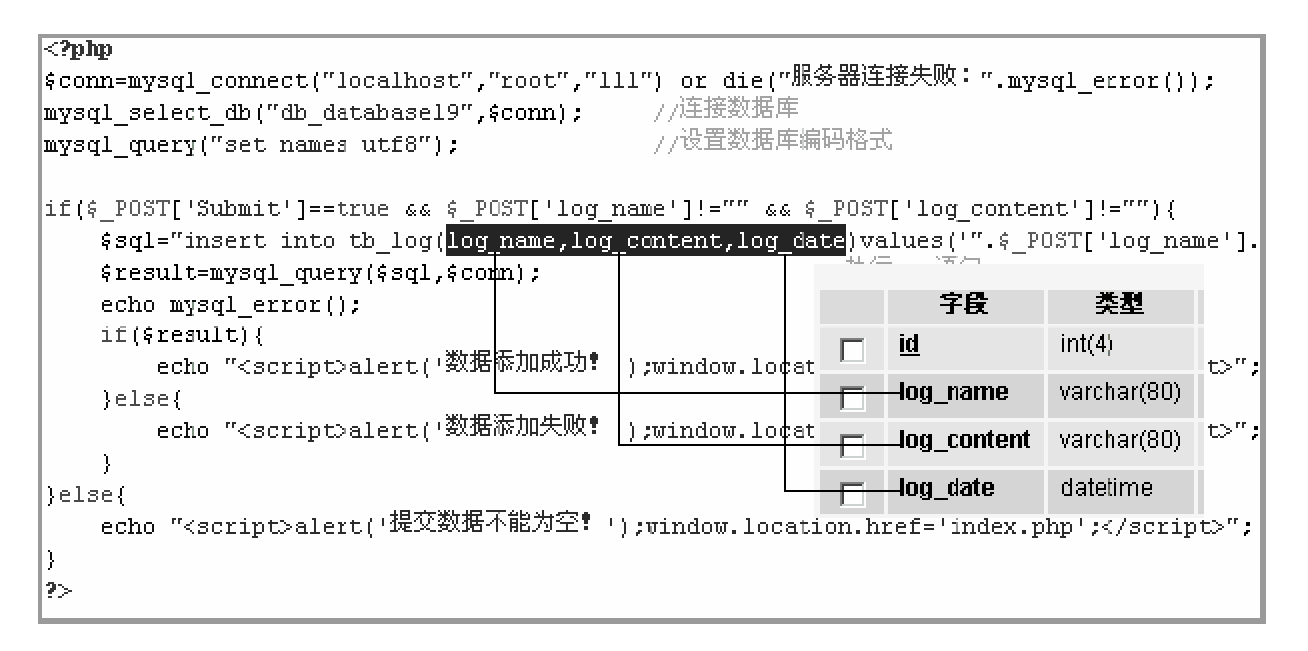
图 21.20　表单元素名称与$_POST[]方法中定义的名称必须相同

第二个环节是数据处理页（index_ok.php 文件）。在该页的 SQL 语句中，字段名称排列的顺序与提交的变量值的顺序要一一对应，否则也会出现错误，即使有些可以进行存储，但是对于数据类型不匹配的字段之间，就不能执行 SQL 语句（如 VARCHAR 和 DATE 之间的类型就不能混淆）。

字段名称与值之间必须对应

```
$sql="insert into tb_log(log_name,log_content,log_date)values('".$_POST['log_name']."','".$_POST['log_ content']."','".date("Y-m-d H:i:s")."')";                //定义 SQL 语句
```

第三个环节是数据处理页和数据库中的数据表之间。在数据处理页编写的 SQL 语句中使用的字段名称一定要与数据库的数据表中的字段名称相同。对应关系如图 21.21 所示。


```
<?php
$conn=mysql_connect("localhost","root","111") or die("服务器连接失败：".mysql_error());
mysql_select_db("db_database19",$conn);       //连接数据库
mysql_query("set names utf8");                //设置数据库编码格式

if($_POST['Submit']==true && $_POST['log_name']!="" && $_POST['log_content']!=""){
    $sql="insert into tb_log(log_name,log_content,log_date)values('".$_POST['log_name'].
    $result=mysql_query($sql,$conn);
    echo mysql_error();
    if($result){
        echo "<script>alert('数据添加成功！');window.locat
    }else{
        echo "<script>alert('数据添加失败！');window.locat
    }
}else{
    echo "<script>alert('提交数据不能为空！');window.location.href='index.php';</script>";
}
?>
```

	字段	类型
☐	id	int(4)
☐	log_name	varchar(80)
☐	log_content	varchar(80)
☐	log_date	datetime

图 21.21　SQL 语句中字段名称与数据表中字段名称必须对应

如果出现上述问题，有两种解决方案。第一种方案：将程序中的 SQL 语句进行复制，在数据库图形化管理工具中的 SQL 语言下执行复制的语句，如果可以执行，说明 SQL 语句是正确的；如果不能执行，说明 SQL 语句本身存在错误，可能是数据表的名称不对、字段类型不符、字段值的变量不正确等，需要及时地对其进行修改。

第二种方案是通过 mysql_error()语句来检测 SQL 语句中的错误。

技巧：在定义表单元素名称和 SQL 语句中字段名称以及数据表中字段名称时，必须保证大小写的统一，如果字符串的大小写不统一，那么 SQL 语句同样不能正确执行。

DIY：对上述实例进行完善，当数据添加成功后，跳转到 main.php 文件中，循环输出数据库中存储的数据。（20 分）（实例位置：光盘\mr\21\qjyy\01_diy）

21.7.2 情景应用 2——解决数据库乱码问题

视频讲解：光盘\mr\21\lx\解决数据库乱码问题.exe

实例位置：光盘\mr\21\qjyy\02

在通过 PHP 对 MySQL 数据库中的数据进行操作的过程中，有时输出 MySQL 数据库中的数据会出现乱码。这是一个让人十分头痛的问题，纠其根源该问题出现的原因是数据库中数据对字符编码格式的限制。由于在创建数据库时使用的编码格式是 utf8，所以在对数据库中的数据进行输出或者添加过程中必须使用相同的编码格式才行，否则就会出现乱码。

在本实例中，通过 while 语句循环输出数据库中的数据，比较在设置数据库编码格式与不设置数据库编码格式时输出数据的不同之处，其运行结果如图 21.22 和图 21.23 所示。

ID	标题	时间
1	数据库服务器问题	2010-10-22 08:31:49
2	数据表问题	2010-10-22 08:32:23
5	复方法	2010-11-06 07:52:57

图 21.22　正常的数据输出结果

ID	标题	时间
1	????????	2010-10-22 08:31:49
2	?????	2010-10-22 08:32:23
5	???	2010-11-06 07:52:57

图 21.23　数据输出乱码

具体步骤如下：

（1）创建数据库连接文件 conn.php，完成与 MySQL 服务器的连接，用户名是 root，密码是 111。然后，连接 db_database21 数据库。最后设置数据库编码格式为 utf8。关键代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or die("服务器连接失败：".mysql_error()); //连接服务器
mysql_select_db("db_database21",$conn) or die ("数据库连接失败：".mysql_error());     //连接数据库
mysql_query("set names utf8");                                                        //设置数据库编码格式
?>
```

（2）创建 index.php 文件。首先通过 include_once()语句包含数据库连接文件，然后定义 SQL 查询语句，最后执行 SQL 语句，并且通过 while 语句完成数据库中数据的循环输出。关键代码如下：

```
<?php
        include_once("conn/conn.php");                    //包含数据库连接文件
        $sql="select * from tb_log ";                     //定义 SQL 语句
        $query=mysql_query($sql,$conn);                   //执行查询操作
        echo mysql_error();                               //返回错误信息，如果存在
        while($myrow=mysql_fetch_array($query)){          //循环输出查询结果
        ?>
```

```
            <tr>
              <td><?php echo $myrow['id']?></td>
              <td><?php echo $myrow['log_name']?></td>
              <td><?php echo $myrow['log_date'];?></td>
            </tr>
        <?php
              }
?>
```

其运行结果如图 21.22 所示。如果不在数据库连接文件中设置编码格式，将“mysql_query("set names utf8");”注释掉，那么再次运行程序时将输出如图 21.23 所示的内容。

DIY：重新设计上述实例的页面，修改为 gb2312 编码，然后通过 mysql_query()函数设置数据库的编码格式为 gb2312，重新输出数据库中的数据。（20 分）（实例位置：光盘\mr\21\qjyy\02_diy）

21.7.3　情景应用 3——封装异常处理类

视频讲解：光盘\mr\21\lx\封装异常处理类.exe

实例位置：光盘\mr\21\qjyy\03

在程序调试中，可以应用 try{}catch{}语句捕获错误信息，并且通过 Exception 内置类来返回相应的错误信息。在此基础上还可以使用继承对 Exception 类进行扩展，编写属于自己的异常处理类。其方法就是，编写一个子类来继承 Exception 类，这样在子类中继承了父类的所有属性和方法，并且还可以添加子类所特有的属性和方法。在本项目中编写一个可以判断电话号码格式是否正确的类，当定义的电话号码格式不正确时跳转到自定义的错误页面，并且输出错误提示信息。运行结果如图 21.24 所示。

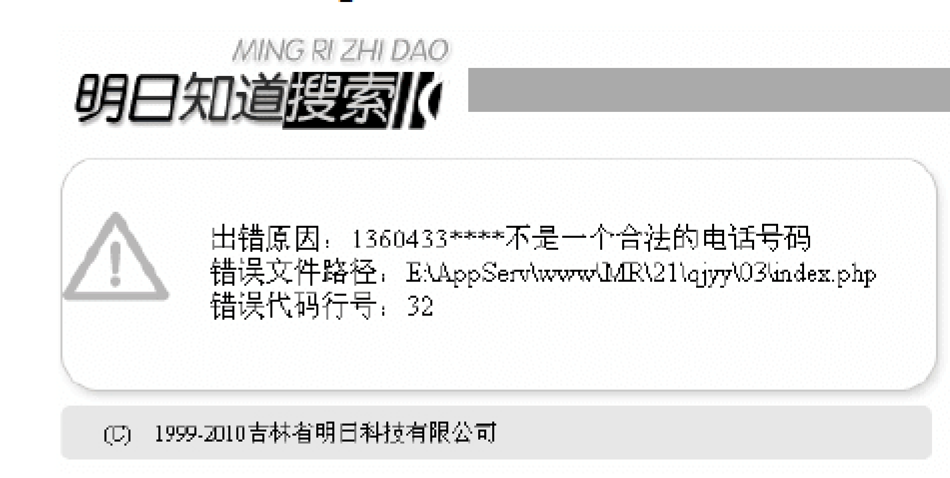


图 21.24　判断电话号码的格式是否正确

本实例的操作步骤如下：

（1）封装电话号码格式判断的异常处理类 TelException，继承 Exception 类。关键代码如下：

```
class TelException extends Exception{                        //定义 TelException 类，继承 Exception 类
          public function errorTel(){                        //定义方法返回错误信息
               $errorMsg = "出错原因：".$this->getMessage()."不是一个合法的电话号码";
               $errorMsg .="<br>";
               $errorMsg .="错误文件路径：".$this->getFile();
               $errorMsg .="<br>";
               $errorMsg .="错误代码行号：".$this-> getLine();
               return $errorMsg;
          }
}
```

（2）创建自定义函数 check_tel()，通过正则表达式和 preg_match()函数验证电话号码的格式是否正确。自定义函数的语法如下：

```
function check_tel($tel){                              //自定义函数验证电话号码格式是否正确
     $checkphone="/^13(\\d{9})$/";                     //定义验证手机号码的正则表达式
     $counts=preg_match($checkphone,$tel);             //执行验证操作
     return $counts;                                   //返回验证结果
}
```

（3）定义被验证的电话号码，应用自定义异常处理类对电话号码格式进行验证。代码如下：

```
$tel = "1360433****";                                   //定义被验证的电话号码
/*
      通过自定义异常处理类返回错误提示
*/
try {
      if(check_tel($tel) !=1){
            throw new TelException($tel);
      }
}catch (TelException $e){
      include_once("error.php");
}
```

DIY：修改上述内容，实现对座机号码格式的验证功能。（20 分）（实例位置：光盘\mr\21\qjyy\03_diy）

21.7.4 情景应用 4——使用错误处理器记录日志

视频讲解：光盘\mr\21\lx\使用错误处理器记录日志.exe

实例位置：光盘\mr\21\qjyy\04

一般情况下，在开发网站时，为了调试方便，将错误直接显示在页面上。但当一个网站正式发布到互联网上时，就不能把内部的错误信息显示给访问者，其最佳的方法应该是记录到错误日志中。本实例实现一个错误处理的例子，将日志记录到日志文件中，并在文件大于 1024B（1KB）时对日志文件进行重命名。本实例的关键代码如下：

```
<?php
function err_log($error,$error_str){                    //自定义一个错误处理函数
      $file="php_errors.log";
      if(filesize($file)>1024){                          //如果日志文件大于 1024B
            rename($file,$file.(string)time());          //以时间为准绳对日志文件进行重命名
            clearstatcache();                            //清除文件状态缓存
      }
      error_log($error_str,3,$file);                     //将出错信息记录到指定的文件中
}
set_error_handler('err_log');                            //执行自定义函数
trigger_error(time().":程序报错.\n");                     //发出错误信息
restore_error_handler();                                 //重新编译这个函数
?>
```

PHP 提供内建的错误处理函数 error_log()，可以将出错信息记录到管理员所指定的路径。error_log()函数的语法如下：

```
error_log(message,type [, destination, [,extra_headers]]);
```

参数 message 指出错信息，参数 type 指定出错信息记录的指定位置。如果用 PHP 的日志记录机制保存出错信息，需要将参数 type 的值设置为 0；如果将错误追加到指定的文件中，需要将参数 type 的值设置为 3。

将错误日志存储到指定文件中，不在页面中显示，还需要对 php.ini 文件做如下设置：

```
display_errors = Off
```

设置完成后，保存 php.ini 文件，然后重新启动 Apache 服务器即可。

DIY：修改上述内容，将错误信息追加到 Apache 服务器的错误日志文件中。提示：将 error_log()函数的第二个参数设置为 0。（20 分）（实例位置：光盘\mr\21\qjyy\04_diy）



21.7.5　情景应用 5——通过 phpMyAdmin 调试 SQL 语句中的错误

视频讲解：光盘\mr\21\lx\通过 phpMyAdmin 调试 SQL 语句中的错误.exe

如果在程序中编写的 SQL 语句不能够正确地执行，那么也可以将 SQL 语句复制到 phpMyAdmin 图形化管理工具中进行调试，在对应的数据表中执行定义的 SQL 语句，看其是否能够正确地运行，如果可以正常运行，那么说明 SQL 语句编写的没有问题，否则可以获取到 SQL 语句的错误信息。

例如，定义一个 select 查询语句“select * from tb_logs”，它在程序中没有正确地执行，然后将其复制到 phpMyAdmin 图形化管理工具中进行调试，其操作步骤如下：

（1）打开 phpMyAdmin 图形化管理工具，进入到指定的数据库，定位到使用的数据表，如图 21.25 所示。

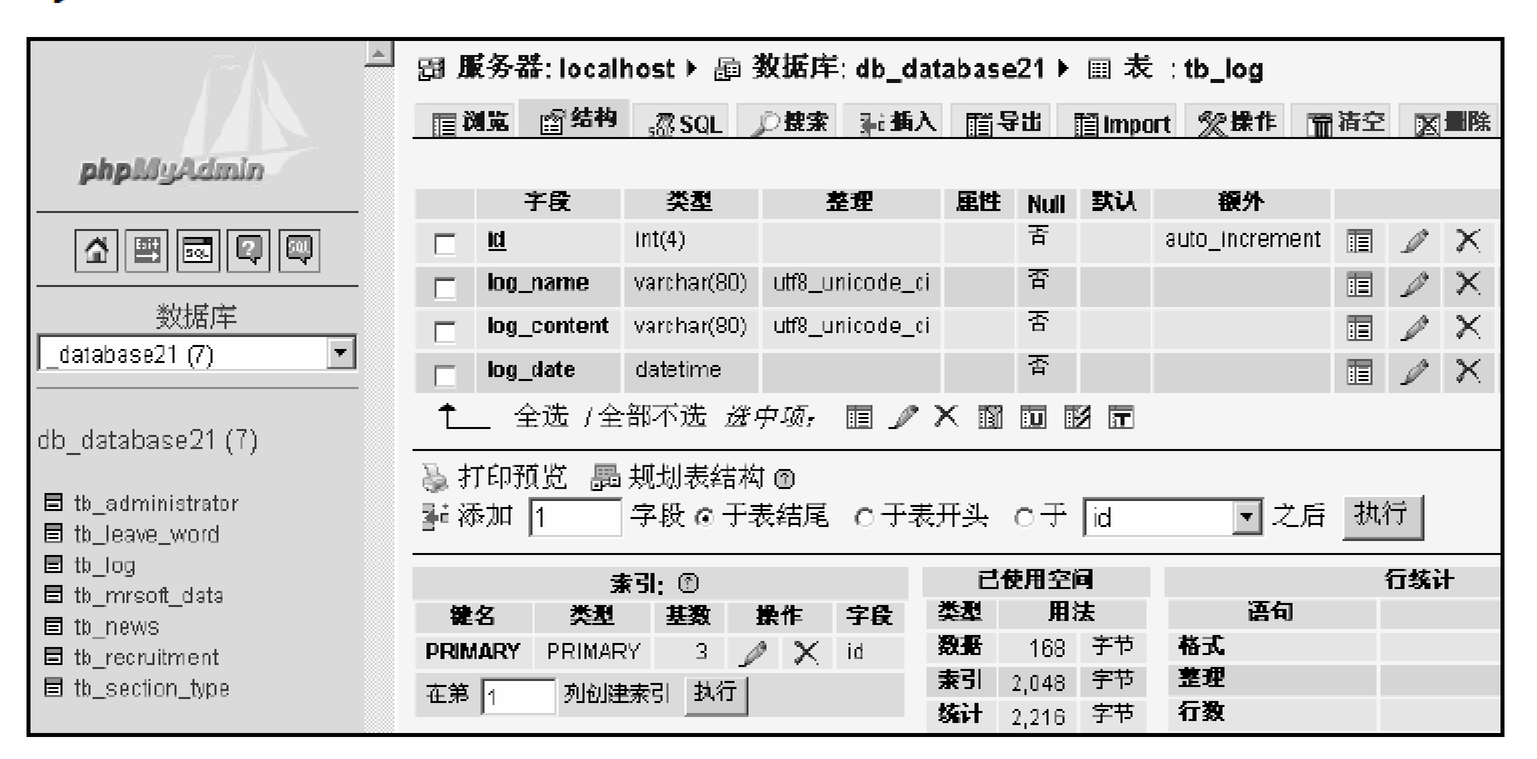


图 21.25　打开 phpMyAdmin 图形化管理工具

（2）单击图 21.25 中的 SQL 按钮，进入到如图 21.26 所示的操作界面，输入要执行的 SQL 语句，单击“执行”按钮。



图 21.26　执行 SQL 语句

（3）如果 SQL 语句没有问题，则执行成功，否则将给出错误提示。

通过这个方法就可以检测出编写的 SQL 语句中是否存在问题。

DIY：在本机的 phpMyAdmin 图形化管理工具中对 SQL 进行调试。（20 分）

情景应用 DIY 栏目分数统计：

DIY 题目	1	2	3	4	5	总分数	
分数							

21.8 自 我 测 试

一、选择题（每题 10 分，5 道题）

1．“echo mrkeji;” 这段代码存在什么错误（　　）。

A．定义错误　　B．语法错误　　C．逻辑错误

2．echo name="mrsoft";，指出这段代码出现错误的原因（　　）。

A．应该用单引号包含变量的值　　B．变量名称缺少$符号

C．变量名称应该大写　　D．缺少分号

3．致命的错误常量值为（　　）。

A．E_ERROR　　B．E_WARNING　　C．E_NOTICE

4．报告通告的错误常量值为（　　）。

A．E_ERROR　　B．E_WARNING　　C．E_NOTICE

5．PHP 的运行环境进行初始化时出现的问题常量值为（　　）。

A．E_ERROR　　B．E_WARNING　　C．E_NOTICE

二、填空题（每题 10 分，5 道题）

1．运用 DIE 语句输出当程序发生错误时提示“FALSE”的代码为（　　　　　　　　）。

2．隐藏错误需要使用特殊符号（　　　　）。

3．利用@屏蔽连接数据库为 db_database21 的操作代码为（　　　　　　　）。

4．控制文件系统是否可以操作远程文件需要修改 php.ini 中的（　　　　）。

5．利用 mysql_error()函数输出连接 MySQL 运行时的错误代码为（　　）。

测试分数统计：

类别	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题
选择题分数					
填空题分数					
总分数					

21.9 行 动 指 南

开始日期：______年____月____日　　　　结束日期：______年____月____日

序号	内　容	行 动 指 南	
1	照猫画虎栏目	分数>75 分	优秀，基本功掌握得不错，加油！
		75 分>分数>50 分	及格，知识掌握得不牢，重新做一遍照猫画虎。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	情景应用栏目	分数>75 分	优秀，综合应用能力很强。
		75 分>分数>50 分	及格，综合应用能力需提高，再练一遍情景应用。
	分数（　　）	分数<50 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。

续表

序号	内　　容	行 动 指 南	
1	自我测试栏目	分数>75 分	优秀，有成为编程高手的潜质。
	分数（　　）	分数<75 分	请使用光盘中提供的“实战能力测试系统”进行提高训练。
	综合评价	反复训练后，以上各项分数都在 75 分以上，方可进入下一堂课学习。	
2	编程能力培养：本栏目提供了一些实践题目，对于培养编程能力很有效，要做好。	（1）在 php.ini 中控制错误处理的级别。	
		（2）封装一个自定义的错误提示信息。	
		（3）通过 mysql_error()函数获取 SQL 语句中出现的错误。	
3	编程习惯培养：本堂课培养读者在学习开发中记笔记的习惯，把开发中遇到的问题、总结的经验记录下来。		
4	创新能力培养：看看身边有没有可以用编程解决的问题，记录到右边的表格中。根据学习进度，尝试编写解决这些问题的小程序。		

21.10　成功可以复制——Java 技术之父 James Gosling

成就：亲手设计了 Java 语言，并开发了 Java 编译器和 Java 虚拟机。
国籍：加拿大
服务公司：IBN、SUN、Oracle
主要职务：SUN 公司副总裁、Oracle 公司客户端软件集团首席技术官
最高学历：博士
出生日期：1955 年

成功历程

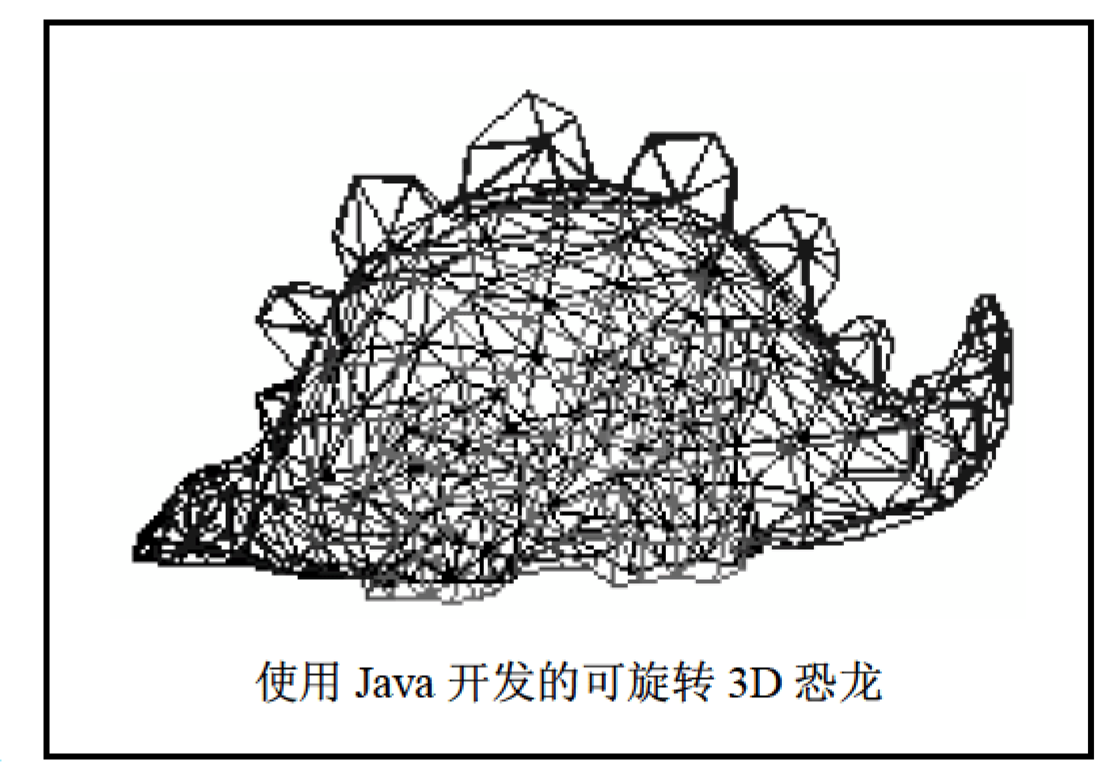
James 出生在加拿大的一个小村庄里，Gosling 家族的第一个男孩，谁也没有想到，James 长大后会成为改变计算机语言的天才。

20 世纪 80 年代初，詹姆斯·戈士林（James Gosling）获得博士学位后到 IBM 工作，设计 IBM 第一代工作站。但是，IBM 的领导层不并看重这个工作站项目。失望之余，James Gosling 跳槽到了 SUN 公司。他花了 5 年时间领导开发的 OS2 很类似于 SUN NeWs 窗口系统，尽管得到了技术界的好评，但该产品并没有得到流行。当时他还是第一个用 C 实现的 EMACC 的所见即所得文本编辑器 COSMACS。

1990 年，SUN 公司成立了一个软件设计团队，其中除了 James Gosling，还有 Patrick Naughton 和 Mike

Sheridan 等人，他们合作的项目称为“绿色计划”。他们认为计算机技术发展的一个趋势是数字家电之间的通信。James 开始负责为设备和用户之间的交流创建一种能够实现网络交互的语言。随着大量时间和金钱的投入，绿色小组创建了一种语言，起初被叫做 Oak，后来被改为 Java。

1994 年底，James Gosling 参加了在硅谷召开的“技术、教育和设计”大会，并向与会者演示了 Java 的功能。James Gosling 单击了网页上的一个静止的分子结构图标，一条命令通过互联网送到了几百英里外的网站，下载了一段 Java 小程序，在本地工作站上开始执行。在几秒钟时间里，原本静止的网页上，一个分子图像以三维动画的方式转了起来。没想到的是 James Gosling 得到了全场观众的赞叹声。



使用 Java 开发的可旋转 3D 恐龙

不久后，硅谷最大的报纸《圣荷西信使报》在头版刊登了一篇专文，“为什么 SUN 认为一杯热咖啡能让你鼓足精神”。从文章见报开始，负责 Java 市场人员的电话响个不停。不久，全美的主要报刊杂志都刊登了 Java 的报道。Java 出名了。James Gosling 先生以“Java 技术之父”而闻名于世。

戈士林先生还曾经创建了卫星数据拦截系统、Linux 的多处理器版本、编译器、邮件系统和视窗管理器等。另外，还创建了著名的 WYSIWYG（所见即所得）测试编译器。

深度评价

也许在 Java 研发时，James Gosling 也没有预想到会取得如此的关注与成功。但是 Java 的成功并非偶然，它的问世凝聚了设计者的辛勤汗水与智慧。也许在 Java 的设计道路上也是荆棘丛生，但是设计者仍然是以勇敢者的气魄，克服种种困难，最终迎接他们的是胜利的曙光。



实战篇

- 第 22 堂课　应用 Smarty 模板开发电子商务网站
- 第 23 堂课　Zend Framework 框架开发明日搜索引擎
- 第 24 堂课　学校图书馆管理系统

第22堂课

应用 Smarty 模板开发电子商务网站

（视频讲解：65 分钟）

随着 20 世纪 PC 机（个人计算机）的发展和互联网的普及，电子商务从报文时代进入到了 Internet 时代，并逐渐被大众所了解和接受。电子商务（Electronic Commerce，EC）是目前发展较快的一种商务模式。迄今为止，不同领域的人对 EC 的理解各有不同。简单地说，EC 是一种基于 Internet，利用计算机硬件、软件等现有设备和协议进行各种商务活动的方式。

学习摘要：

- 开发背景
- 需求分析
- 系统分析
- 数据库设计
- 公共文件设计
- 前台首页设计
- 登录模块设计
- 会员信息模块设计
- 商品展示模块设计
- 购物车模块设计
- 收银台模块设计
- 后台首页设计

22.1 开 发 背 景

自 20 世纪 90 年代，互联网的蓬勃发展，为企业提供了一个全新的机遇。企业网站、电子商务成为热门话题。其中，电子商务更是关系到经济结构、产业升级和国家整体经济竞争力。为此，我国已经将发展电子商务列为信息化建设的重要内容，并努力创造条件，积极地推进电子商务的发展。

据美国在线（AOL）和 Henley Centre 联合进行的一项调查显示：国外有 80%的受调查者会选择网上购物或寻求帮助，10%的受调查者会选择熟悉的品牌或厂商来购买。而在国内，自 1997 年拉开了电子商务的序幕，短短的 10 年时间里，全国已有 4 万家商业网站，几乎每天都有新的网站诞生，厂商所在地也从上海、广州、深圳等沿海发达地区扩展到全国各大中城市。

22.2 需 求 分 析

随着“地球村”概念的兴起，网络已经深入到人们生活的每一个角落。世界越来越小，信息的传播越来越快，内容也越来越丰富。现在，人们对于在网络上寻求信息和服务已不再满足于简单的信息获取上，人们更多的是需要在网上实现便捷的、可交互式的网络服务。电子商务则正好满足了人们的需求。它可以让人们在网上实现互动的交流及足不出户地购买产品，向企业发表自己的意见、服务需求及有关投诉，并且通过网站的交互式操作向企业进行产品的咨询、得到相应的回馈及技术支持。精明的商家绝不会错过这样庞大的市场，越来越多的企业已经开展了电子商务活动。加入电子商务的行列也许不会让企业马上见到效益，但不加入则一定会被时代所抛弃。

22.3 系 统 分 析

22.3.1 系统目标

根据客户提供的需求和对实际情况的考察与分析，该电子商务应该具备如下特点：

- ☑ 首页设计要能够吸引用户的目光，整个页面要以简洁为主，突出重点。
- ☑ 可操作性强，避免复杂的、有异议的链接。
- ☑ 浏览速度快，尽量避免长时间打不开页面的情况发生。
- ☑ 商品信息部分有实物图例，图像清楚、文字醒目。
- ☑ 详细的商品查询功能，可以通过商品的各个属性来搜索。
- ☑ 详细的流程介绍，从浏览商品到购买结账，各个步骤之间的联系最好能以图例来说明。
- ☑ 提供在线咨询。
- ☑ 后台可以对用户信息和商品信息进行详尽的查看和管理。
- ☑ 订单管理。
- ☑ 易维护，并提供二次开发支持。



22.3.2　系统功能结构

电子商务平台分前台系统和后台系统。下面分别给出前、后台的系统功能结构图。电子商务前台系统功能结构如图 22.1 所示。

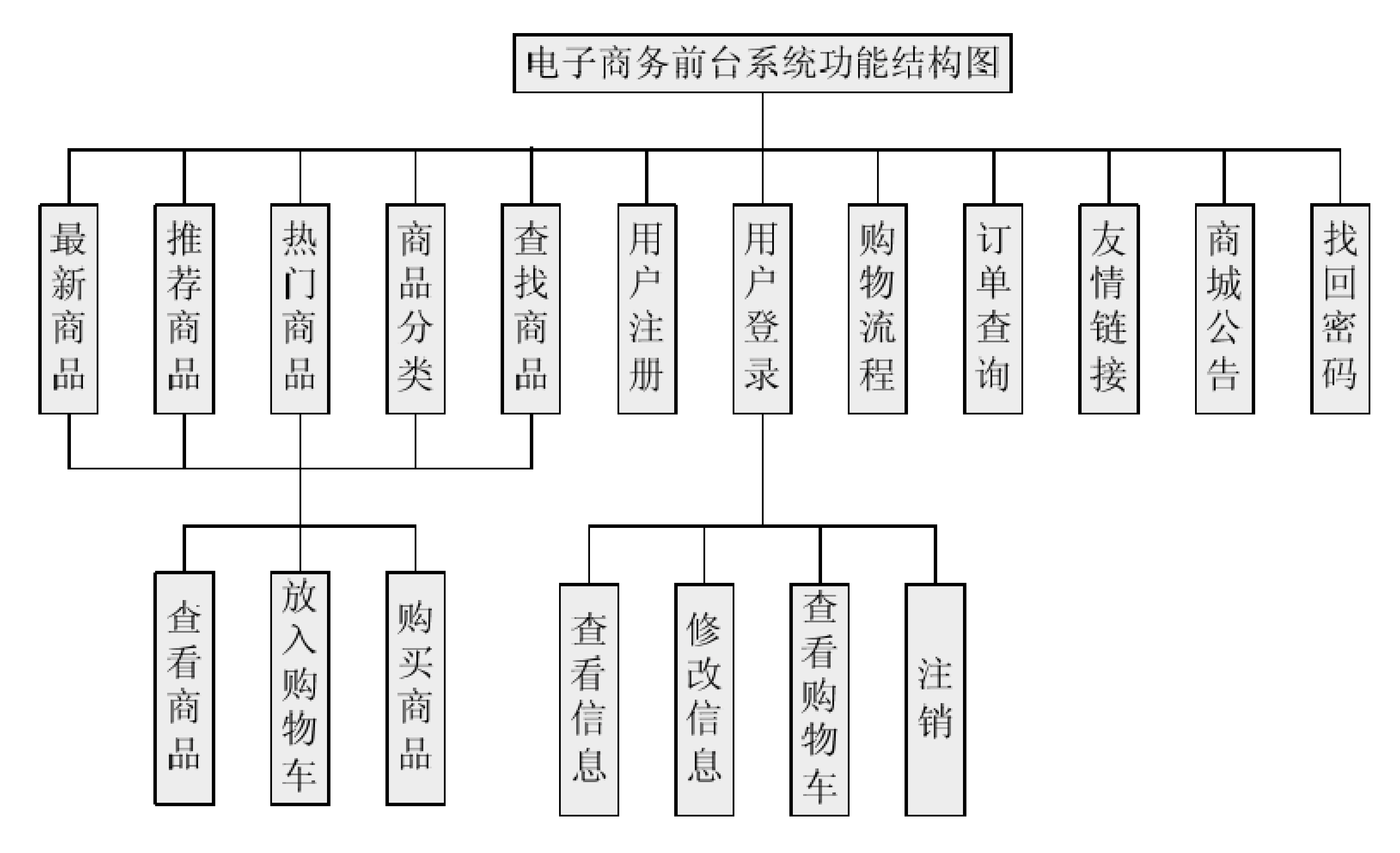


图 22.1　电子商务前台系统功能结构图

电子商务后台系统功能结构图如图 22.2 所示。

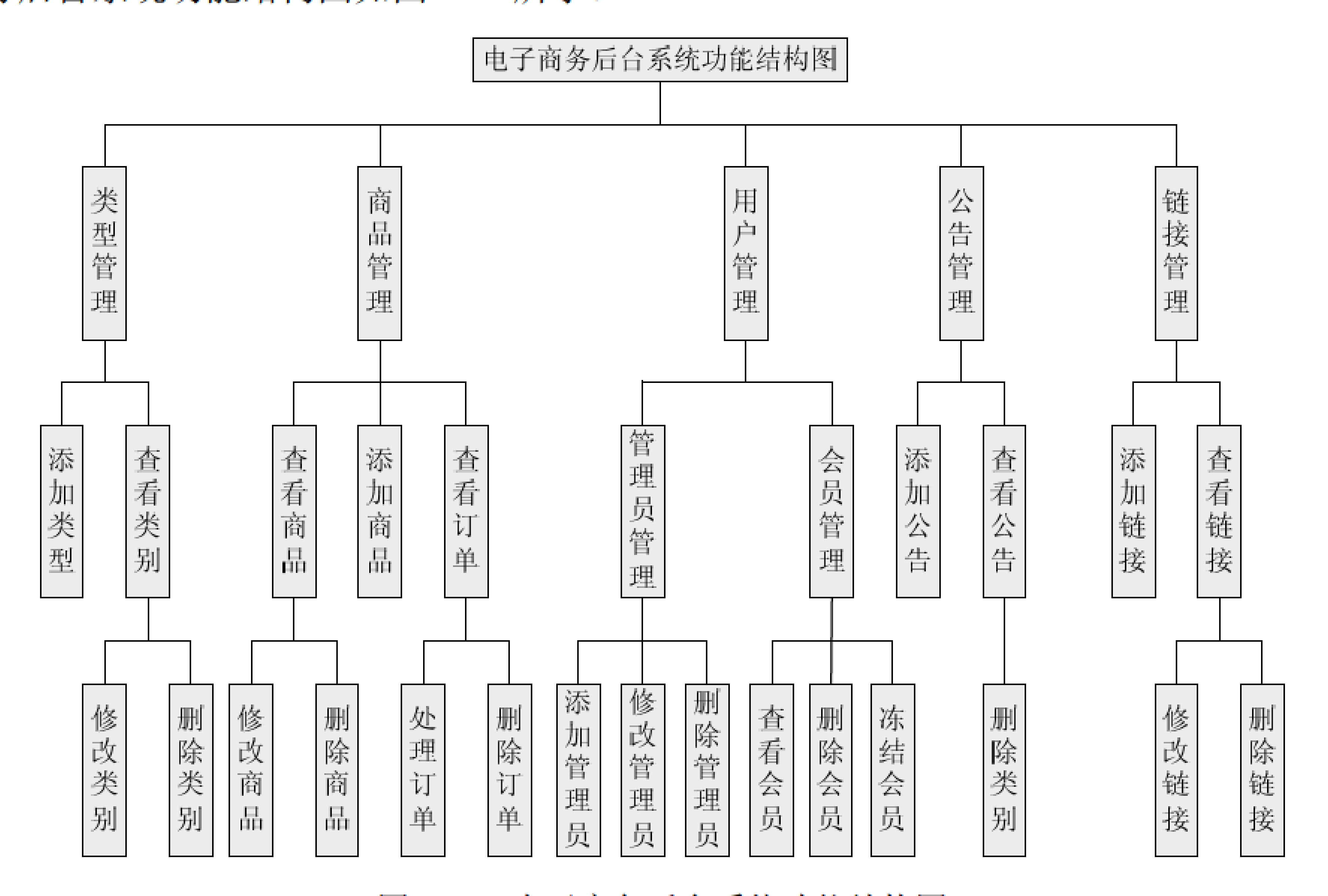


图 22.2　电子商务后台系统功能结构图

22.3.3　开发环境

在开发电子商务平台时，该项目使用的软件开发环境介绍如下。

1．服务器端

- ☑ 操作系统：Windows Server 2003/Linux（推荐）。
- ☑ 服务器：Apache 2.2.8。
- ☑ PHP 软件：PHP 5.2.6。
- ☑ 数据库：MySQL 5.0.51。
- ☑ MySQL 图形化管理软件：phpMyAdmin-2.22.3。
- ☑ PDO 数据库抽象层。
- ☑ Smarty 模板。
- ☑ 开发工具：Dreamweaver 8。
- ☑ 浏览器：IE 6.0 及以上版本。
- ☑ 分辨率：最佳效果为 1024 像素×768 像素。

2．客户端

- ☑ 浏览器：推荐 IE 6.0 及以上版本。
- ☑ 分辨率：最佳效果为 1024 像素×768 像素。

22.3.4 文件夹组织结构

编写代码之前，可以把系统中可能用到的文件夹先创建出来（例如，创建一个名为 images 的文件夹，用于保存程序中所使用的图片），这样不但可以方便以后的开发工作，也可以规范系统的整体架构。因为本项目使用的是 Smarty+PDO 技术，所以目录较多。下面介绍本系统的目录结构（到三级目录），如图 22.3 所示。

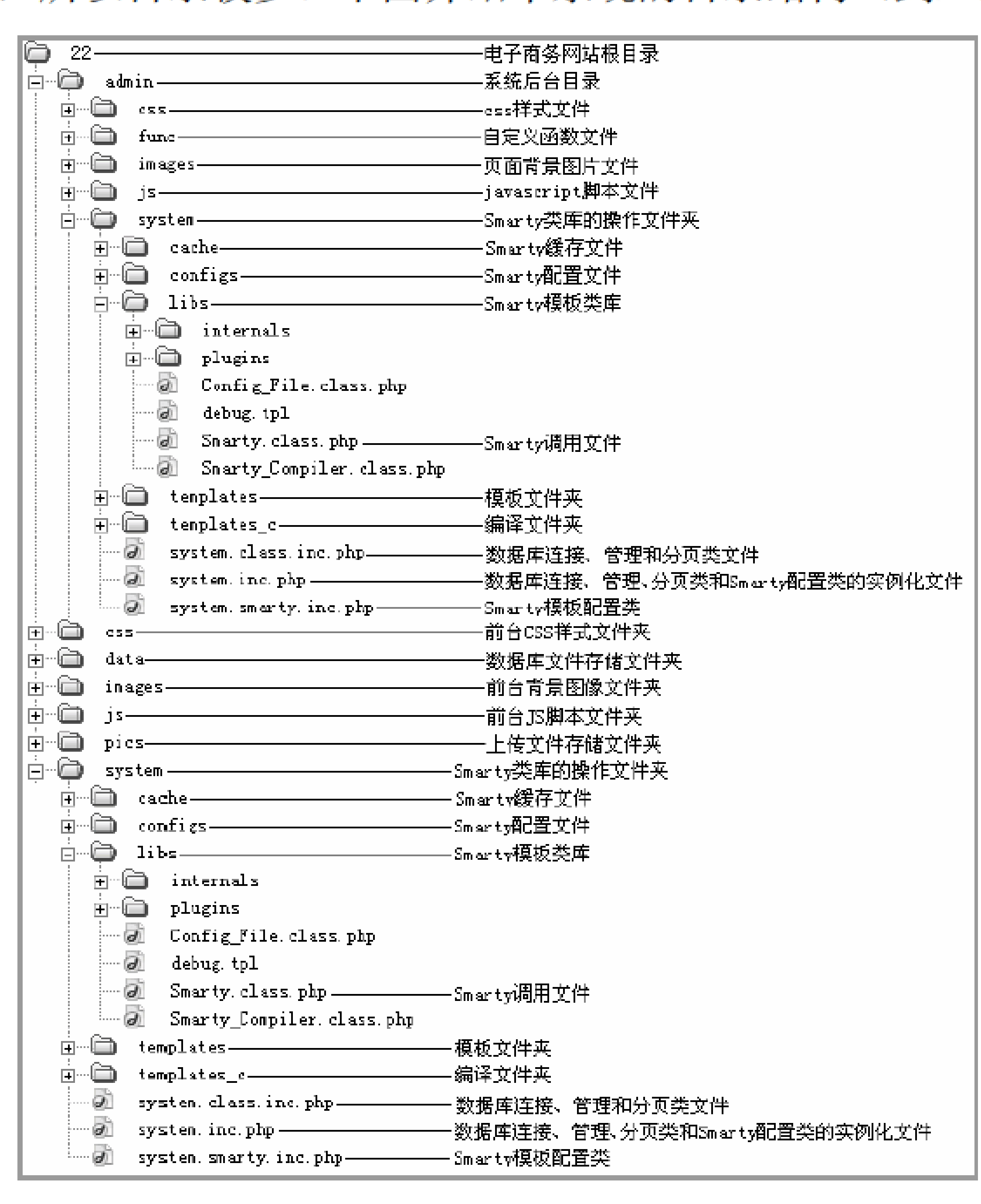


图 22.3 电子商务网站文件夹组织结构图

22.4　数据库设计

无论是什么系统软件，其最根本的功能就是对数据的操作与使用。所以，一定要先做好数据的分析、设计与实现，然后再实现对应的功能模块。

22.4.1　数据库分析

根据需求分析和系统的功能流程图，找出需要保存的信息数据（也可以理解为现实世界中的实体），并将其转化为原始数据（属性类型）形式。这种描述现实世界的概念模型，可以使用 E-R 图来表示，也就是实体-联系图。最后将 E-R 图转换为关系数据库。这里重点介绍几个 E-R 图。

1．会员信息实体

会员信息实体包括编号、用户名、密码、E-mail、身份证号、联系电话、QQ 号、密码提示、密码答案、邮编、注册时间、真实姓名等属性。会员信息实体 E-R 图如图 22.4 所示。

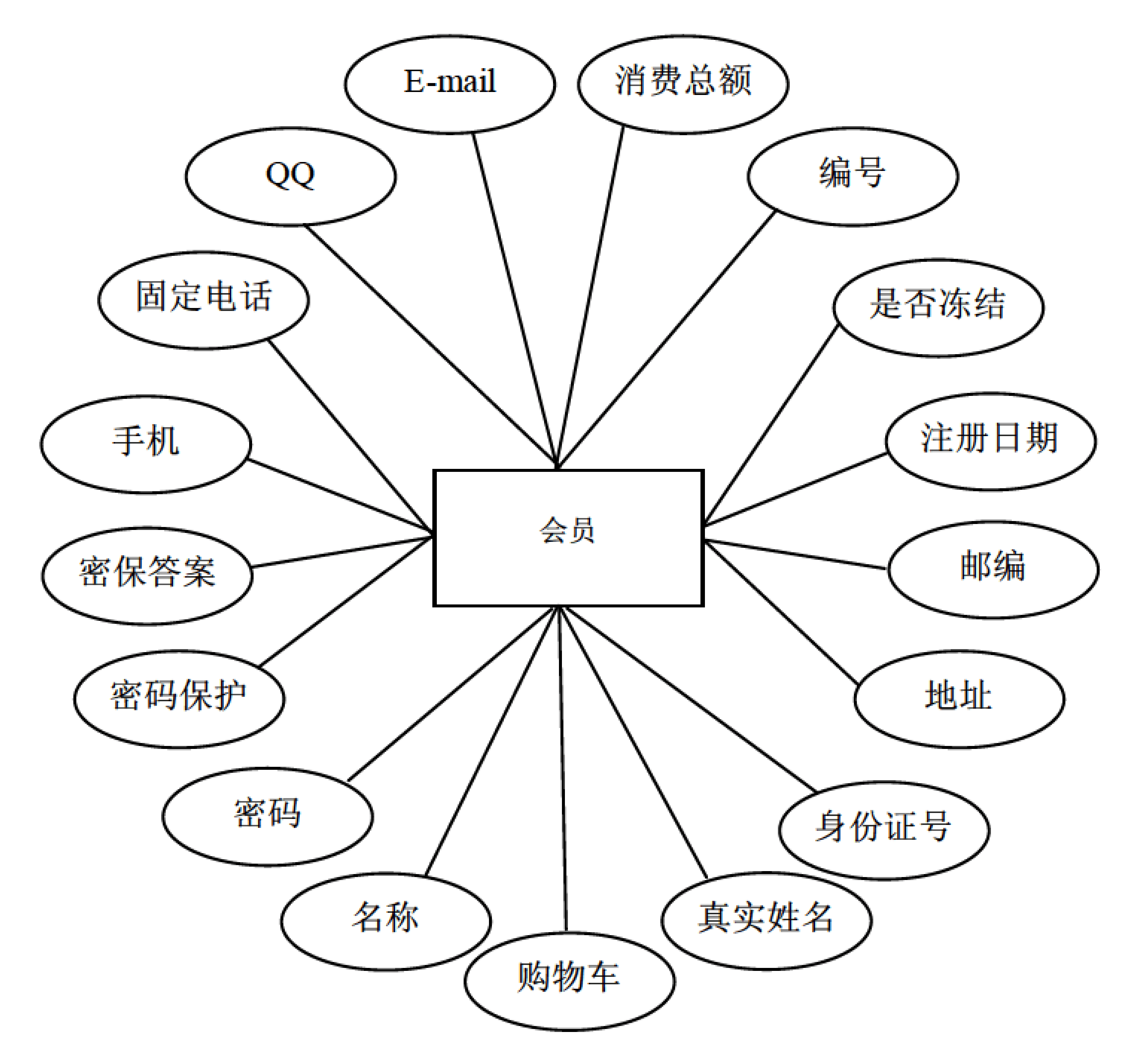


图 22.4　会员信息实体 E-R 图

2．商品信息实体

商品信息实体包括编号、名称、上市时间、添加时间、型号、图片、库存、销售、商品类型、会员价、市场价、是否打折等属性。商品信息实体 E-R 图如图 22.5 所示。

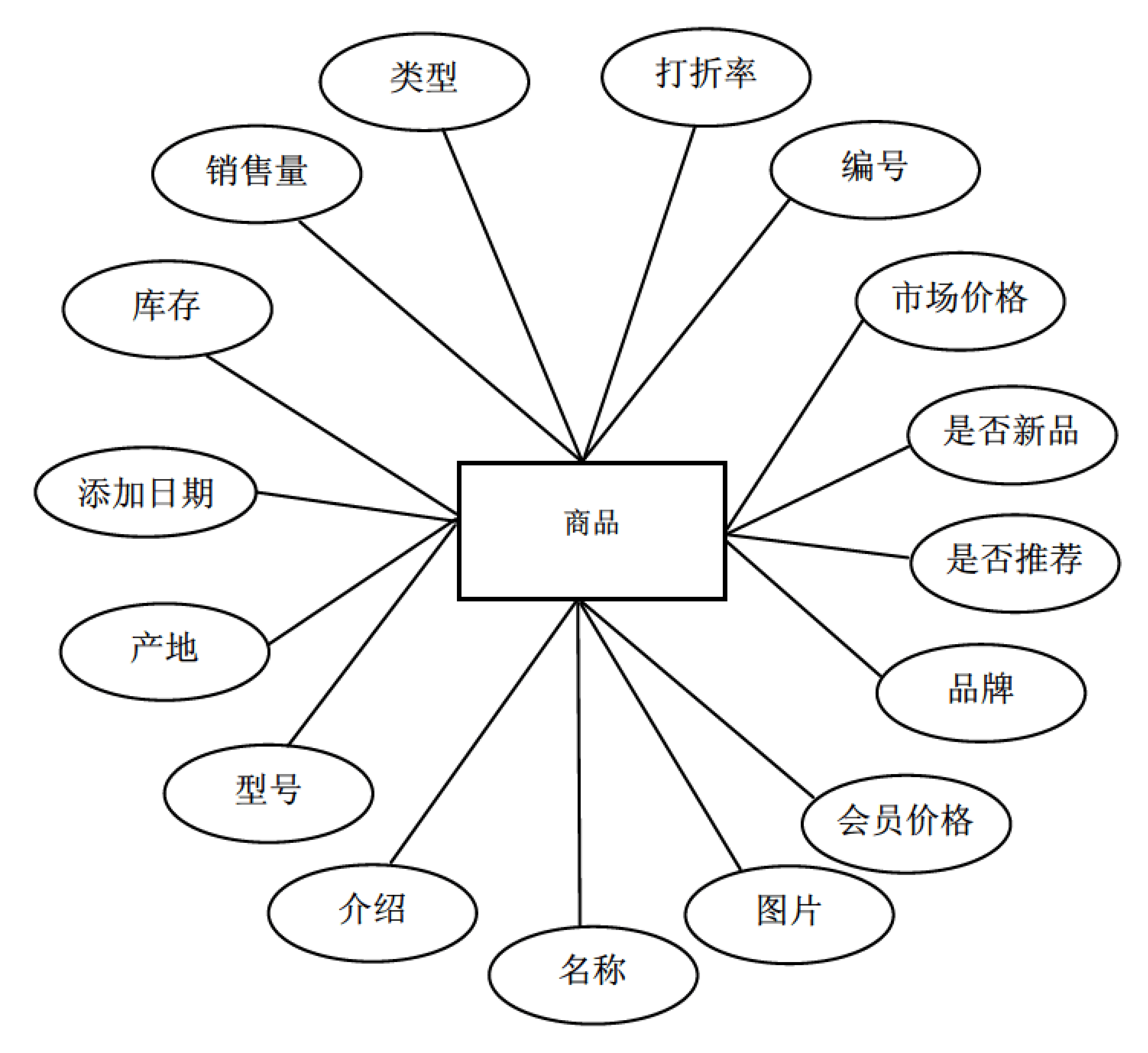


图 22.5　商品信息实体 E-R 图

除上面介绍的两个 E-R 图，还有商品订单实体、商品评价实体、公告实体、管理员实体、类型实体和友情链接实体等。限于篇幅，这里仅列出主要的实体 E-R 图。

22.4.2　创建数据库和数据表

系统 E-R 图设计完成后，接下来根据 E-R 图来创建数据库和数据表。首先来看电子商务平台所使用的数据表情况，如图 22.6 所示。

下面来看各个数据表的结构和字段说明。

☑　tb_admin（管理员信息表）

管理员信息表主要用于存储管理员的信息，其结构如图 22.7 所示。

表	操作						记录数	类型	整理
tb_admin							1	MyISAM	utf8_unicode_ci
tb_class							6	MyISAM	utf8_unicode_ci
tb_commo							6	MyISAM	utf8_unicode_ci
tb_form							14	MyISAM	utf8_unicode_ci
tb_links							3	MyISAM	utf8_unicode_ci
tb_opinion							0	MyISAM	utf8_unicode_ci
tb_public							5	MyISAM	utf8_unicode_ci
tb_user							3	MyISAM	utf8_unicode_ci

图 22.6　电子商务数据表



字段	类型	整理	属性	Null	默认	额外
id	int(4)			否		auto_increment
name	varchar(50)	utf8_unicode_ci		否		
pwd	varchar(50)	utf8_unicode_ci		否		

图 22.7　管理员信息表结构

☑　tb_class（商品类型列表）

商品类型列表主要用于添加商品的类别，可以设定多个子类别（目前最多只能到二级子类别），其结构如图 22.8 所示。



☑ tb_commo（商品信息表）

商品信息表主要用于存储关于商品的相关信息，其结构如图 22.9 所示。

字段	类型	整理	属性	Null	默认	额外
id	int(4)			否		auto_increment
name	varchar(20)	utf8_unicode_ci		否		
supid	int(2)			否		

图 22.8 商品类型列表结构

字段	类型	整理	属性	Null	默认	额外
id	int(4)			否		auto_increment
name	varchar(50)	utf8_unicode_ci		否		
pics	varchar(200)	utf8_unicode_ci		否	pics/null.jpg	
info	mediumtext	utf8_unicode_ci		否		
addtime	date			否		
area	varchar(50)	utf8_unicode_ci		否		
model	varchar(50)	utf8_unicode_ci		否		
class	varchar(50)	utf8_unicode_ci		否		
brand	varchar(50)	utf8_unicode_ci		否		
stocks	int(4)			否	1	
sell	int(4)			否	0	
m_price	float			否		
v_price	float			否		
fold	float			否	9	
isnew	int(1)			否	1	
isnom	int(1)			否	0	

图 22.9 商品信息表结构

此外还有商品订单表、商品公告表、用户信息表、友情链接表和商品留言表。限于篇幅，这里不再介绍，读者可参见本书附赠光盘中的数据库文件。

22.5 公共文件设计

公共模块就是将多个页面都可能使用到的代码写成单独的文件，在使用时只要用 include 或 require 语句将文件包含进来即可。如本系统中的数据库连接、管理和分页类文件，Smarty 模板配置类文件，执行类的实例化文件，CSS 样式表文件，js 脚本文件等，以前台系统为例，下面给出主要的公共文件，后台的公共文件与前台大同小异。

22.5.1 数据库连接、管理和分页类文件

在数据库连接、管理和分页类文件中定义 3 个类，分别是 ConnDB 数据库连接类，实现通过 PDO 连接 MySQL 数据库；AdminDB 数据库管理类，使用 PDO 类库中的方法执行对数据库中数据的查询、添加、更新和删除操作；SepPage 分页类用于对商城中的数据进行分页输出。

代码位置：光盘\MR\22\system\system.class.inc.php

```
<?php
//数据库连接类
class ConnDB{
        var $dbtype;
        var $host;
      var $user;
      var $pwd;
      var $dbname;
       //构造方法
      function ConnDB($dbtype,$host,$user,$pwd,$dbname){
```

```
                $this->dbtype=$dbtype;
                $this->host=$host;
                $this->user=$user;
                $this->pwd=$pwd;
                $this->dbname=$dbname;
        }
        //实现数据库的连接并返回连接对象
        function GetConnId(){
         if($this->dbtype=="mysql" || $this->dbtype=="mssql"){
                        $dsn="$this->dbtype:host=$this->host;dbname=$this->dbname";
                }else{
                        $dsn="$this->dbtype:dbname=$this->dbname";
                }
                try {
                        $conn = new PDO($dsn, $this->user, $this->pwd);
                                                        //初始化一个 PDO 对象，就是创建了数据库连接对象$pdo
                        $conn->query("set names utf8");
                        return $conn;
                } catch (PDOException $e) {
                        die ("Error!: " . $e->getMessage() . "<br/>");
                }
        }
}
//数据库管理类
class AdminDB{
        function ExecSQL($sqlstr,$conn){
                $sqltype=strtolower(substr(trim($sqlstr),0,6));
                $rs=$conn->prepare($sqlstr);                    //准备查询语句
                $rs->execute();                                 //执行查询语句，并返回结果集
                if($sqltype=="select"){
                        $array=$rs->fetchAll(PDO::FETCH_ASSOC);         //获取结果集中的所有数据
                        if(count($array)==0 || $rs==false)
                                return false;
                        else
                                return $array;
                }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
                        if($rs)
                                return true;
                        else
                                return false;
                }
        }
}
//分页类
class SepPage{
        var $rs;
        var $pagesize;
        var $nowpage;
        var $array;
        var $conn;
        var $sqlstr;
        function ShowData($sqlstr,$conn,$pagesize,$nowpage){  //定义方法
```

```
		if(!isset($nowpage) || $nowpage=="")			//判断变量值是否为空
			$this->nowpage=1;					//定义每页起始页
		else
			$this->nowpage=$nowpage;
		$this->pagesize=$pagesize;					//定义每页输出的记录数
		$this->conn=$conn;						//连接数据库返回的标识
		$this->sqlstr=$sqlstr;						//执行的查询语句
		$this->rs=$this->conn->PageExecute($this->sqlstr,$this->pagesize,$this->nowpage);
		@$this->array=$this->rs->GetRows();				//获取记录数
			if(count($this->array)==0 || $this->rs==false)
				return false;
			else
				return $this->array;
	}
	function ShowPage($contentname,$utits,$anothersearchstr,$anothersearchstrs,$class){
		$allrs=$this->conn->Execute($this->sqlstr);			//执行查询语句
		$record=count($allrs->GetRows());				//统计记录总数
		$pagecount=ceil($record/$this->pagesize);			//计算共有几页
		$str.=$contentname." ".$record." ".$utits." 每页 ".$this->pagesize."  ".
$utits." 第 ".$this->rs->AbsolutePage()." 页/共 ".$pagecount." 页";
		$str.="    ";
		if(!$this->rs->AtFirstPage())
			$str.="<a  href=".$_SERVER['PHP_SELF']."?page=1&parameter1=".$anothersearchstr.  "&parameter2=".
$anothersearchstrs." class=".$class.">首页</a>";
		else
			$str.="<font color='#555555'>首页</font>";
		$str.=" ";
		if(!$this->rs->AtFirstPage())
			$str.="<a href=".$_SERVER['PHP_SELF']."?page=".($this->rs->AbsolutePage()-1)."&parameter1= ".
$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">上一页</a>";
		else
			$str.="<font color='#555555'>上一页</font>";
		$str.=" ";
		if(!$this->rs->AtLastPage())
			$str.="<a href=".$_SERVER['PHP_SELF']."?page=".($this->rs->AbsolutePage()+1)."&parameter1 =".
$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">下一页</a>";
		else
			$str.="<font color='#555555'>下一页</font>";
		$str.=" ";
		if(!$this->rs->AtLastPage())
			$str.="<a href=".$_SERVER['PHP_SELF']."?page=".$pagecount."&parameter1=".$anothersearchstr."
&parameter2=".$anothersearchstrs." class=".$class.">尾页</a>";
		else
			$str.="<font color='#555555'>尾页</font>";
		if(count($this->array)==0 || $this->rs==false)
			return "";
		else
			return $str;
	}
}
?>
```

22.5.2 Smarty 模板配置类文件

在 Smarty 模板配置类文件中配置 Smarty 模板文件、编译文件、配置文件等文件路径。

代码位置：光盘\MR\22\system\system.smarty.inc.php

```
<?php
require("libs/Smarty.class.php");                           //包含模板文件
class SmartyProject extends   Smarty{                       //定义类，继承模板类
      function SmartyProject(){                             //定义方法
            $this->template_dir = "./system/templates/";    //指定模板文件存储位置
            $this->compile_dir = "./system/templates_c/";   //指定编译文件存储位置
            $this->config_dir = "./system/configs/";        //指定配置文件存储位置
            $this->cache_dir = "./system/cache/";           //指定缓存文件存储位置
      }
}
?>
```

22.5.3 执行类的实例化文件

在 system.inc.php 文件中，通过 require 语句包含 system.smarty.inc.php 和 system.class.inc.php 文件，执行类的实例化操作，并定义返回对象。完成数据库连接类的实例化后，调用其中的 GetConnId()方法连接数据库。

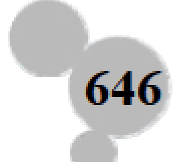
代码位置：光盘\MR\22\system\system.inc.php

```
<?php
require("system.smarty.inc.php");                                              //包含 Smarty 配置类
require("system.class.inc.php");                                               //包含数据库连接和操作类
$connobj=new ConnDB("mysql","localhost","root","111","db_database22");         //数据库连接类实例化
$conn=$connobj->GetConnId();                                                   //执行连接操作，返回连接标识
$admindb=new AdminDB();                                                        //数据库操作类实例化
$seppage=new SepPage();                                                        //分页类实例化
$usefun=new UseFun();                                                          //使用常用函数类实例化
$smarty=new SmartyProject();                                                   //调用 Smarty 模板
function unhtml($params){
      extract($params);
      $text=$content;
      global $usefun;
      return $usefun->UnHtml($text);
}
$smarty->register_function("unhtml","unhtml");                                 //注册模板函数
?>
```

22.6 前台首页设计

22.6.1 前台首页概述

前台首页一般没有多少实质的技术，主要是加载一些功能模块，如登录模块、导航栏模块、公告栏模块等，使浏览者能够了解网站内容和特点。首页的重要之处是要合理地对页面进行布局，既要尽可能地将



重点模块显示出来，同时又不能因为页面凌乱无序，而让浏览者无所适从、产生反感。本系统前台首页的运行结果如图 22.10 所示。



图 22.10　前台首页运行结果

22.6.2　前台首页技术分析

在前台首页中应用 switch 语句与 Smarty 模板中的内建函数 include 设计一个框架页面，实现不同功能模块在首页中的展示。

switch 语句在 PHP 动态文件中使用，根据超链接传递的值，包含不同的功能模块。

include 标签在 Smarty 模板页中使用，在当前模板页中包含其他模板文件。其语法如下：

```
{include file="file_name " assign=" " var="   "}
```

其中，参数 file 指定包含模板文件的名称；assign 指定一个变量保存包含模板的输出；var 传递给待包含模板的本地参数，只在待包含模板中有效。

22.6.3　前台首页实现过程

前台首页实现过程的步骤如下：

（1）创建 index.php 动态页。在 index.php 动态页中，应用 include_once()语句包含相应的文件；应用 switch 语句，以超链接中参数 page 传递的值为条件进行判断，实现在不同页面之间跳转。index.php 的关键代码如下：

代码位置：光盘\MR\22\index.php

```
<?php
session_start();
```

```
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
require("system/system.inc.php");                                  //包含配置文件
if(isset($_GET["page"])){
    $page=$_GET["page"];
}else{
    $page="";
}
include_once("login.php");
include_once("public.php");
include_once("links.php");
switch($page){
    case "hyzx":
        include_once "member.php";
        $smarty->assign('admin_phtml','member.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    case 'allpub':
        include_once 'allpub.php';
        $smarty->assign('admin_phtml','allpub.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    case 'nom':
        include_once 'allnom.php';
        $smarty->assign('admin_phtml','allnom.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    case 'new':
        include_once 'allnew.php';
        $smarty->assign('admin_phtml','allnew.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    case 'hot':
        include_once 'allhot.php';
        $smarty->assign('admin_phtml','allhot.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    case 'shopcar':
        include_once 'myshopcar.php';
        $smarty->assign('admin_phtml','myshopcar.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    case 'settle':
        include_once 'settle.php';
        $smarty->assign('admin_phtml','settle.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    case 'queryform':
        include_once 'queryform.php';
        $smarty->assign('admin_phtml','queryform.tpl');//将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
    default:
        include_once 'newhot.php';
        $smarty->assign('admin_phtml','newhot.tpl');   //将 PHP 脚本文件对应的模板文件名称赋给模板变量
    break;
}
$smarty->display("index.tpl");                        //指定模板页
?>
```

（2）创建 index.tpl 模板页。在模板文件 index.tpl 中应用 Smarty 的 include 标签调用不同的模板文件，生成静态页面。关键代码如下：

代码位置：光盘\MR\22\system\templates\index.tpl

```
<table width="850" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td colspan="2">{include file='top.tpl'}</td>
  </tr>
  <tr>
    <td width="216" align="left" valign="top">
     {include file='login.tpl'}
     {include file='public.tpl'}
     {include file='links.tpl'}
    </td>
    <td width="634" height="700" align="center" valign="top">
{include file='search.tpl'}
<!--载入模板文件-->{include file=$admin_phtml}</td>
  </tr>
</table>
<table width="850" border="0" cellspacing="0" cellpadding="0">
     <tr>
     <td>{include file='buttom.tpl'}</td>
    </tr>
</table>
```

说明： 本系统的功能较多，结构比较复杂，对于初学者来说学起来可能会比较困难。所以，本书将系统中的各个功能模块所涉及的文件（如 PHP、TPL、CSS、JS 等）尽可能都单独实现。读者在学习其中某个模块时，可以将相关的文件统一放到同一个目录下单独测试。

22.7　登录模块设计

22.7.1　登录模块概述

用户登录模块是会员功能的窗口。匿名用户虽然也可以访问本网站，但只能进行浏览、查询等简单操作，而会员则可以购买商品，并且能享受超低价格。登录模块包括用户注册、用户登录和找回密码 3 部分，其运行结果如图 22.11 所示。



图 22.11　登录模块运行效果

22.7.2　登录模块技术分析

（1）Ajax 技术无刷新验证用户名是否被占用。关键代码如下：

代码位置：光盘\MR\22\js\check.js

```
/*  form 为传入的表单名称，本段代码为 register 表单  */
function chkname(form){
	/*  如果 name 文本域的信息为空，名为 name1 的 div 标签显示如下信息  */
	if(form.name.value==""){
		name1.innerHMRL="<font color=#FF0000>请输入用户名！</font>";
	}else{
		/*  否则获取文本域的值  */
		var user = form.name.value;
		/*  生成 url 链接，将 user 的值传到 chkname.php 页进行判断  */
		var url = "chkname.php?user="+user;
		/*  使用 xmlhttprequest 技术运行页面  */
		xmlhttp.open("GET",url,true);
		xmlhttp.onreadystatechange = function(){
		if(xmlhttp.readyState == 4){
				/*  根据不同的返回值，在 div 标签中输出不同信息  */
				var msg = xmlhttp.responseText;
				if(msg == '3'){
					name1.innerHMRL="<font color=#FF0000>用户名被占用！</font>";
					return false;
				}else if(msg == '2'){
					name1.innerHMRL="<font color=green>恭喜您，可以注册!</font>";
				/*  如果用户名正确，则将隐藏域的值改为“yes”  */
					form.c_name.value = "yes";
				}else{
					name1.innerHMRL="<font color=green>未知错误</font>";
				}
			}
		}
		xmlhttp.send(null);
	}
}
```

在该函数中调用 chkname.php 页，该页在会员登录时也会被调用，所以这里分两种情况，有密码和无密码。无密码为注册验证，当没有返回结果时，说明该用户名可用；而有密码为登录验证，和无密码相反，只有查询记录存在时，才允许登录，并将用户名和用户 ID 存储到 session 中。chkname.php 页面的代码如下：

代码位置：光盘\MR\22\chkname.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );		//设置文件编码格式
require("system/system.inc.php");				//包含配置文件
$reback = '0';
$sql = "select * from tb_user where name='".$_GET['user']."'";
if(isset($_GET['password'])){
	$sql .= " and password = '".md5($_GET['password'])."'";
```

```
}
$rst = $admindb->ExecSQL($sql,$conn);
if($rst){
	/*   登录所用   */
	if($rst[0]['isfreeze'] != 0){
		$reback = '3';
	}else{
		$_SESSION['member'] = $rst[0]['name'];
		$_SESSION['id'] = $rst[0]['id'];
		$reback = '2';
	}
}else{
	$reback = '1';
}
echo $reback;
?>
```

（2）GD2 函数库生成验证码，关键代码如下：

代码位置：光盘\MR\22\yzm.php

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
srand((double)microtime()*1000000);                                  //生成随机数
$im=imagecreate(60,30);                                              //创建画布
$black=imagecolorallocate($im,0,0,0);                                //定义背景
$white=imagecolorallocate($im,255,255,255);                          //定义背景
$gray=imagecolorallocate($im,200,200,200);                           //定义背景
imagefill($im,0,0,$gray);                                            //填充颜色
for($i=0;$i<4;$i++){                                                 //定义 4 位随机数
  $str=mt_rand(3,20);                                                //定义随机字符所在位置的 Y 坐标
  $size=mt_rand(5,8);                                                //定义随机字符的字体
  $authnum=substr($_GET['num'],$i,1);                                //获取超链接中传递的验证码
  imagestring($im,$size,(2+$i*15),$str,$authnum,imagecolorallocate($im,rand(0,130),rand(0,130),rand(0,130)));
}                                                                    //水平输出字符串
for($i=0;$i<200;$i++){                                               //执行 for 循环，为验证码添加模糊背景
   $randcolor=imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255));    //创建背景
   imagesetpixel($im,rand()%70,rand()%30,$randcolor);                //绘制单一元素
}
imagepng($im);                                                       //生成 png 图像
imagedestroy($im);                                                   //销毁图像
?>
```

22.7.3　用户注册

用户注册页面的主要功能是新用户注册。如果信息输入完整而且符合要求，则系统会将该用户的信息保存到数据库中，否则显示错误原因，以便用户改正。用户注册页面的运行结果如图 22.12 所示。

用户注册的步骤如下：

（1）创建 register.tpl 模板文件，编写用户注册页面。其中包含两个 JS 脚本文件 createxmlhttp.js 和 check.js。其中，createxmlhttp.js 是 Ajax 的实例化文件，而 check.js 对用户注册信息进行验证，并且返回验证结果。

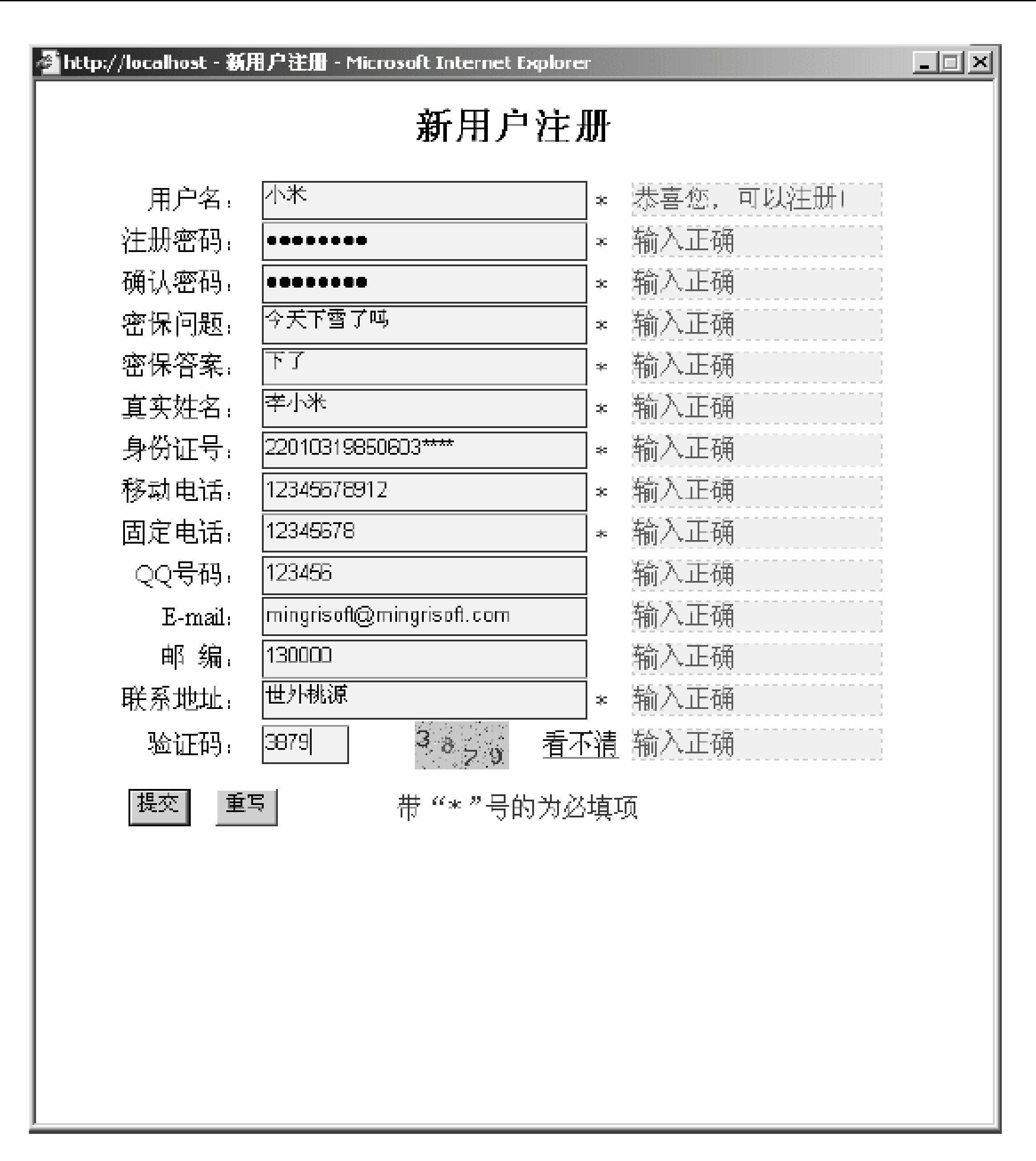


图 22.12　注册模块页面

（2）创建 register.php 动态 PHP 文件，加载模板。register.php 文件的代码如下：

代码位置：光盘\MR\22\register.php

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
require("system/system.inc.php");                             //包含配置文件
$smarty->assign('title','新用户注册');
$smarty->display('register.tpl');
?>
```

（3）创建 reg_chk.php 文件，获取表单中提交的数据，将数据存储到指定的数据表中。reg_chk.php 文件的代码如下：

代码位置：光盘\MR\22\reg_chk.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
require("system/system.inc.php");                             //包含配置文件
    $name = $_POST['name'];
    $password = md5($_POST['pwd1']);
    $question = $_POST['question'];
    $answer = $_POST['answer'];
    $realname = $_POST['realname'];
    $card = $_POST['card'];
    $tel = $_POST['tel'];
    $phone = $_POST['phone'];
    $Email = $_POST['email'];
    $QQ = $_POST['qq'];
    $code = $_POST['code'];
    $address = $_POST['address'];
```

```
	$addtime = date("Y-m-d H:i:s");
	$sql = "insert into tb_user(name,password,question,answer,realname,card,tel,phone,Email,QQ,code,
address,addtime,isfreeze,shopping)" ;
	$sql .= " values ('$name', '$password', '$question', '$answer', '$realname', '$card', '$tel', '$phone', '$Email',
'$QQ', '$code', '$address','$addtime','0','')";
	$rst= $admindb->ExecSQL($sql,$conn);			//执行添加操作
	if($rst){
		$_SESSION['member'] = $name;
		echo "<script>top.opener.location.reload();alert('注册成功');window.close();</script>";
	}else{
		echo '<script>alert(\'添加失败\');history.back;</script>';
	}
?>
```

（4）创建“用户注册”超链接。当用户单击前台的注册按钮时，系统会调用 js 的 onClick 事件，弹出注册窗口。代码如下：

代码位置：光盘\MR\22\system\templates\login.tpl

```
<a href="#" id="login" onClick="reg()"><img src="images/check.JPG" width="59" height="23" border="0" /></a>
```

这里使用到的 js 文件为 js\login.js，调用的函数为 reg()。该函数的代码如下：

代码位置：光盘\MR\22\js\login.js

```
function reg(){
window.open("register.php", "_blank", "width=600,height=650",false);		//弹出窗口
}
```

22.7.4　用户登录

用户登录模块的运行结果如图 22.13 所示，需要输入用户名、密码和验证码。

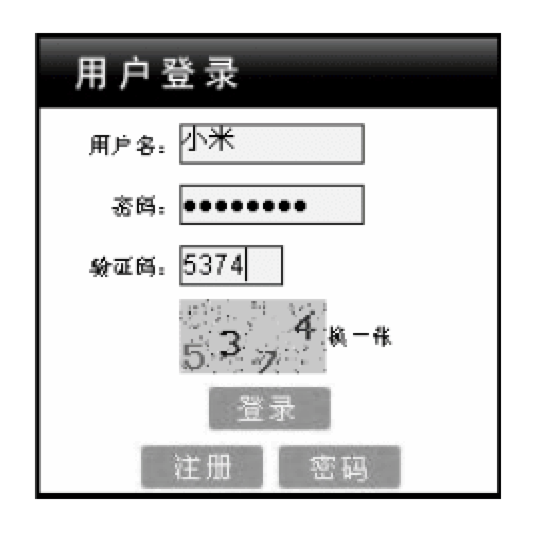


图 22.13　用户登录页面

用户登录的步骤如下：

（1）创建模板文件 login.tpl，完成用户登录表单的设计。在该页面中当单击“登录”按钮时，系统将调用 lg()函数对用户登录提交信息进行验证。lg()函数包含在 js\login.js 脚本文件内，代码如下：

代码位置：光盘\MR\22\js\login.js

```
// JavaScript Document
function lg(form){
	if(form.name.value==""){
		alert('请输入用户名');
		form.name.focus();
		return false;
	}
```

```
		if(form.password.value == "" || form.password.value.length < 6){
			alert('请输入正确密码');
			form.password.focus();
			return false;
		}
		if(form.check.value == ""){
			alert('请输入验证码');
			form.check.focus();
			return false;
		}
		if(form.check.value != form.check2.value){
			form.check.select();
			code(form);
			return false;
		}
		var user = form.name.value;
		var password = form.password.value;
		var url = "chkname.php?user="+user+"&password="+password;
		xmlhttp.open("GET",url,true);
		xmlhttp.onreadystatechange = function(){
		if(xmlhttp.readyState == 4){
					var msg = xmlhttp.responseText;
					if(msg == '1'){
						alert('用户名或密码错误!!');
						form.password.select();
						form.check.value = '';
						code(form);
						return false;
					}if(msg == "3"){
						alert("该用户被冻结，请联系管理员");
						return false;
					}else{
						alert('欢迎光临');
						location.reload();
					}
			}
		}
		xmlhttp.send(null);
		return false;
}
//显示验证码
function yzm(form){
		var num1=Math.round(Math.random()*10000000);
		var num=num1.toString().substr(0,4);
		document.write("<img name=codeimg width=65 heigh=35 src='yzm.php?num="+num+"'>");
		form.check2.value=num;
}
//刷新验证码
function code(form){
		var num1=Math.round(Math.random()*10000000);
```

```
        var num=num1.toString().substr(0,4);
        document.codeimg.src="yzm.php?num="+num;
        form.check2.value=num;
}
//注册
function reg(){
window.open("register.php", "_blank", "width=600,height=650",false);
}
//找回密码
function found() {
window.open("found.php","_blank","width=350 height=240",false);
}
```

用户名和密码是在 chkname.php 页面中被验证。chkname.php 在 22.7.2 节中已经介绍，这里不再重复。

（2）创建用户信息模板文件 info.tpl。用户登录成功后，在原登录框位置将显示用户信息，用户可以通过“会员中心”对自己的信息做修改，也可以单击“查看购物车”超链接查看购物车商品；当用户离开时可以单击“安全离开”超链接。用户信息模块的主要代码如下：

代码位置：光盘\MR\22\system\templates\info.tpl

```
<!--  显示当前登录用户名  -->
欢迎您：{$member}
<!--  会员中心超链接  -->
<a href="?page=hyzx" id="info" class="lk">会员中心</a>
<!--  查看购物车  -->
<a href="?page=shopcar" class="lk">查看购物车</a>
<!--  安全离开  -->
<a onclick="javascript:logout()" style="cursor:hand" id="info">安全离开</a>
```

22.7.5　找回密码

登录模块的最后一个部分就是找回密码。找回密码是根据用户在填写资料时所填写的密保问题和密保答案来实现的。当用户单击“找回密码”超链接时，首先提示用户输入要找回密码的会员名称，然后根据密保问题填写密保答案，最后重新输入密码。找回密码模块的流程如图 22.14 所示。



图 22.14　找回密码流程图

1．创建模板文件

虽然找回密码需要 4 个步骤，但实际上每个步骤使用的都是相同的模板文件和 js 文件，只是被调用的表单和 js 函数略有差别。这里根据不同的文件来分别进行介绍。

该模板文件一共包含了 3 个表单，分别代表了 3 个步骤，其核心代码如下：

代码位置：光盘\MR\22\system\templates\found.tpl

```
<!-- 载入两个 js 脚本文件 -->
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/found.js"></script>
<!-- 第一个 div 标签 -->
<div id="first">
<table width="200" border="0" cellspacing="0" cellpadding="0">
<form id="foundname" name="found" method="post" action="#">
  <tr><td> 找回密码</td></tr>
  <tr><td>会员名称：</td>
     <!-- text 文本域，用于输入要找回密码的会员名称 -->
<td><input id="user" name="user" type="text" class="txt"></td>
</tr>
  <tr><td>
<!-- 单击“下一步”按钮，能触发 onClick 事件来调用 chkname()函数 -->
<input id = " next1 " name = " next1 " type = " button " class = " btn " value = " 下一步 " onClick = " return chkname ( foundname ) "/></td></tr>
</form>
</table>
</div>
<!-- 第二个 div 标签，样式为隐藏 -->
<div id="second" style="display:none;">
<table>
<form id="foundanswer" name="found" method="post" action="#">
  <tr><td > 找回密码</td></tr>
  <tr><td>密保问题：</td>
<!-- 用于显示密保问题的 div 标签 -->
    <td <div id="question"></div></td></tr>
  <tr><td>密保答案：</td>
<!-- 文本域，用于填写密保答案 -->
    <td ><input id="answer" name="answer" type="text" class="txt" /></td></tr>
  <tr>
<!-- 单击“下一步”按钮，用来触发 onClick 事件，并调用 chkanswer()函数 -->
    <td><input id = " next2 " name = " next2 " type=" button " class=" btn " value =" 下一步 " onClick = " return chkanswer ( foundanswer ) "></td>
    </tr>
</form>
</table>
</div>
<!-- 第 3 个 div 标签，样式也为隐藏，作用是修改密码 -->
<div id='third' style="display:none;">
<table>
<form id="modifypwd" name="found" method="post" action="#">
  <tr><td> 输入密码</td></tr>
```

```
  <tr><td>输入密码：</td>
    <td><input id="pwd1" name="pwd1" type="password" class="txt"></td></tr>
  <tr><td>确认密码：</td>
    <td><input id="pwd2" name="pwd2" type="password" class="txt" /></td>
  </tr>
  <tr>
<!--  单击“完成”按钮，调用 chkpwd()函数  -->
    <td><input id = " mod " name = " mod " type = " button " class = " btn " value = "  完成  " onClick = " return
chkpwd (modifypwd) "></td>
    </tr>
</form>
</table>
</div>
```

可以看出，在上述 3 个表单中，只有一个表单默认情况下是显示的，其他则为隐藏。只有通过调用不同的 js 函数，才可以对其他表单进行操作。

2．创建 js 脚本文件

found.js 脚本文件包含 3 个函数，即 chkname()、chkanswer()和 chkpwd()。其中，chkname()函数的作用是检查用户输入的会员名称，如果存在，则使用 xmlhttp 对象去调用生成的 url 进行处理判断。如果该用户存在，则隐藏当前表单，并显示下一个表单，最后输出密保问题。chkname()函数的代码如下：

代码位置：光盘\MR\22\js\found.js

```
function chkname(form){
    var user = form.user.value;
    if(user == ''){
        alert('请输入用户名');
        form.user.focus();
        return false;
    }else{
        var url = "foundpwd.php?user="+user;
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
        if(xmlhttp.readyState == 4){
                var msg = xmlhttp.responseText;
                if(msg == '0'){
                    alert('没有该用户，请重新查找!');
                    form.user.select();
                    return false;
                }else{
                    document.getElementById('first').style.display = 'none';
                    document.getElementById('second').style.display = '';
                    document.getElementById('question').innerHTML = msg;
                }
            }
        }
        xmlhttp.send(null);
    }
}
```

其他两个函数也使用 xmlhttprequest 对象，实现方法相差无几，不同之处就是对返回值的处理，chkanswer()

函数隐藏当前表单，显示下一个表单。chkanswer()函数的代码如下：

代码位置：光盘\MR\22\js\found.js

```
function chkanswer(form) {
	var user = document.getElementById('user').value;
	var answer = form.answer.value;
	if(answer == ''){
		alert('请输入提示问题');
		form.answer.focus();
		return false;
	}else{
		var url = "foundpwd.php?user="+user+"&answer="+answer;
		xmlhttp.open("GET",url,true);
		xmlhttp.onreadystatechange = function(){
			if(xmlhttp.readyState == 4){
				var msg = xmlhttp.responseText;
				if(msg == '0'){
					alert('问题回答错误');
					form.answer.select();
					return false;
				}else{
					document.getElementById('second').style.display = 'none';
					document.getElementById('third').style.display = '';
				}
			}
		}
		xmlhttp.send(null);
	}
}
```

而 chkpwd()函数则提示用户操作状态，如果成功，则关闭当前页。chkpwd()函数的代码如下：

代码位置：光盘\MR\22\js\found.js

```
function chkpwd(form){
	var user = document.getElementById('user').value;
	var pwd1 = form.pwd1.value;
	var pwd2 = form.pwd2.value;
	if(pwd1 == ''){
		alert('请输入密码');
		form.pwd1.focus();
		return false;
	}
	if(pwd1.length < 6){
		alert('密码输入错误');
		form.pwd1.focus();
		return false;
	}
	if(pwd1 != pwd2){
		alert('两次密码不相等');
		form.pwd2.select();
		return false;
	}
	var url = "foundpwd.php?user="+user+"&password="+pwd1;
```

```
        xmlhttp.open("GET",url,true);
        xmlhttp.onreadystatechange = function(){
                if(xmlhttp.readyState == 4){
                        var msg = xmlhttp.responseText;
                        if(msg == '1'){
                                alert('密码修改成功，请重新登录');
                                window.close();
                        }else{
                                alert(msg);
                        }
                }
        }
        xmlhttp.send(null);
}
```

3．创建数据处理文件

foundpwd.php 文件的功能是根据用户输入信息来检测数据表中的数据，并根据不同的输入信息返回不同的结果。该文件的代码如下：

代码位置：光盘\MR\22\foundpwd.php

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
require("system/system.inc.php");                                   //包含配置文件
$smarty->assign('title','找回密码');
$reback = '0';                                                      //设置变量初始值
if(!isset($_GET['answer']) && !isset($_GET['password'])){           //判断变量是否存在
        $namesql = "select * from tb_user where name = '".$_GET['user']."'";
        $namerst = $admindb->ExecSQL($namesql,$conn);               //查询用户名是否存在
        if($namerst){
                $question = $namerst[0]['question'];
                $reback = $question;
        }
}else if(isset($_GET['answer'])){
        $answersql = "select * from tb_user where name = '".$_GET['user']."' and answer = '".$_GET['answer']."'";
        $answerrst = $admindb->ExecSQL($answersql,$conn);
        if($answerrst){
                $reback = '1';
        }
}else if(isset($_GET['password'])){
        $sql="update tb_user set password='".md5($_GET['password'])."' where name='".$_GET['user']."'";
        $rst = $admindb->ExecSQL($sql,$conn);
        if($rst){
                $reback = '1';                                      //为模板变量赋值
        }
}
echo $reback;                                                       //输出返回结果
?>
```

4．加载模板页

因为所有登录模块的模板都不需要或者只需要传递一两个变量，所以 PHP 加载页的内容比较简单。找

回密码页面的代码如下：

代码位置：光盘\MR\22\found.php

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
require("system/system.inc.php");                              //包含配置文件
$smarty->assign('title','找回密码');
$smarty->display('found.tpl');
?>
```

22.8　会员信息模块设计

22.8.1　会员信息模块概述

用户登录后，即可看到会员信息模块。在这里，可以查看或修改个人信息及密码、查看购物车和安全退出等操作。本节只对会员信息模块中的“会员中心”和“安全退出”进行讲解，关于“查看购物车”将在商品模块中进行介绍。会员信息模块的运行效果如图 22.15 所示。

mr信息（不可更改信息）	
会员编号：	2
会员名称：	mr
密保问题：	123456
密保答案：	123456
注册时间：	2010-12-06 05:14:10
消费总额：	0
mr信息（可更改信息）	
真实姓名：	潘琴 *
身份证号：	220322821103653 *
移动电话：	13604338784 *
固定电话：	0431-84972266 *
Email：	pp@sina.com
QQ号：	346827366
邮编：	130025
地址：	长春市 *
修改　重置	

图 22.15　会员信息模块

22.8.2　会员信息模块技术分析

在会员信息模块中，以 SESSION 变量中存储的用户名称为条件，从会员信息表中查询出会员信息，并且将会员信息存储到模板变量中，最后在模板页中输出会员信息。member.php 的代码如下：

代码位置：光盘\MR\22\member.php

```
<?php
/*    查找用户资料    */
if(isset($_SESSION['member'])){
    $sql = "select * from tb_user where name = '".$_SESSION['member']."'";
    $arr = $admindb->ExecSQL($sql,$conn);
    if(isset($_GET['action']) && $_GET['action'] == 'modify'){
```



```
            $smarty->assign('check',"find");
            $smarty->assign('pwdarr',$arr);
        }else{
            $smarty->assign('check',"notfind");
            $smarty->assign('pwdarr',$arr);
        }
}
?>
```

member.php 文件中查询出的数据是会员信息模板功能实现的根本。

22.8.3　会员中心

当单击“会员中心”超链接时，会回传给当前页一个 page 值，当前页根据这个 page 值来载入 member.php 文件。

1．创建 PHP 页面

与登录模块设计不同，本节首先来创建 PHP 页面。因为该模块中的模板需要使用数据库中的数据及一些动态信息，这些都需要在 PHP 页中先行获取及处理，然后再传给模板页。会员中心页面的代码请参考 22.8.2 小节的内容。

2．创建模板页

该模块包括查看信息模板及修改密码模板，都存储于 member.tpl 模板文件中。

代码位置：光盘\MR\22\system\templates\member.tpl

```
<link rel="stylesheet" href="css/member.css" />
<script language="javascript" src="js/member.js"></script>
{if $check=="find" }
<p  align="left">{$smarty.session.member}&gt;&gt;&gt;<a  href='?page=hyzx'  id="mem">查看信息</a>&gt;&gt;
&gt;<a href='?page=hyzx&action=modify' id="mem">修改密码</a></p>
<table   id="member" width="300" border="0" cellpadding="0" cellspacing="0">
<form   id="member"   name="member"   method="post"   action="modify_pwd_chk.php"   onSubmit="return
pwd(member)">
  <tr>
    <td  height="25"  colspan="2"  align="center"  valign="middle"  id="first"><font  color="#f0f0f0">修改密码
</font></td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left">原密码：</td>
    <td height="25" align="left" valign="middle" id="right"><input id="old" name="old" type="password" /></td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left">新密码：</td>
    <td  height="25"  align="left"  valign="middle"  id="right"><input  id="new1"  name="new1"  type="password"
/></td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left">确认密码：</td>
    <td  height="25"  align="left"  valign="middle"  id="right"><input  id="new2"  name="new2"  type="password"
/></td>
```

```
  </tr>
  <tr>
    <td height="30" colspan="2" align="center" valign="middle"><input id="enter" name="enter" type="submit"
value="修改" /></td>
  </tr>
</form>
</table>
{else}
<p  align="left">{$smarty.session.member}&gt;&gt;&gt;<a  href='?page=hyzx'  id="mem">查看信息</a>&gt;&gt;
&gt;<a href='?page=hyzx&action=modify' id="mem">修改密码</a></p>
{section name=pwd_id loop=$pwdarr}
<table id='member' width="500" border="0" cellpadding="0" cellspacing="0">
<form   id="member"   name="member"   method="post"   action="modify_info_chk.php"   onSubmit="return
mem(member)" >
  <tr>
    <td height="25" colspan="2" align="center" valign="middle" id="first"><font color="#f0f0f0">{$pwdarr[pwd_
id].name}信息（不可更改信息）</font></td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left"> 会员编号：</td>
    <td height="25" align="left" valign="middle" id="right"> {$pwdarr[pwd_id].id}</td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left"> 会员名称：</td>
    <td height="25" align="left" valign="middle" id="right"> {$pwdarr[pwd_id].name}</td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left"> 密保问题：</td>
    <td height="25" align="left" valign="middle" id="right"> {$pwdarr[pwd_id].question}</td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left">密保答案：</td>
    <td height="25" align="left" valign="middle" id="right"> {$pwdarr[pwd_id].answer}</td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left"> 注册时间：</td>
    <td height="25" align="left" valign="middle" id="right"> {$pwdarr[pwd_id].addtime}</td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left">消费总额：</td>
    <td height="25" align="left" valign="middle" id="right"> {$pwdarr[pwd_id].consume}</td>
  </tr>
  <tr>
    <td height="25" colspan="2" align="center" valign="middle" id="first"><font color="#f0f0f0">{$pwdarr[pwd_
id].name}信息（可更改信息）</font></td>
  </tr>
  <tr>
    <td width="25%" height="25" align="right" valign="middle" id="left">真实姓名：</td>
    <td height="25" align="left" valign="middle" id="right"><input id="realname" name="realname" type="text"
value="{$pwdarr[pwd_id].realname}" /> 
```

```
            <input type="hidden" name="userid" value="{$pwdarr[pwd_id].id}" />
            <font color="red">*</font></td>
    </tr>
    <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left">身份证号：</td>
      <td height="25" align="left" valign="middle" id="right"><input id="card" name="card" type="text" value=
"{$pwdarr[pwd_id].card}" /> <font color="red">*</font></td>
    </tr>
    <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left">移动电话：</td>
      <td height="25" align="left" valign="middle" id="right"><input id="tel" name="tel" type="text" value="{$pwdarr
[pwd_id].tel}"> <font color="red">*</font> </td>
    </tr>
    <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left">固定电话：</td>
      <td height="25" align="left" valign="middle" id="right"><input id="phone" name="phone" type="text"
value="{$pwdarr[pwd_id].phone}" /> <font color="red">*</font></td>
    </tr>
    <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left">Email：</td>
      <td height="25" align="left" valign="middle" id="right"><input id="email" name="email" type="text"
value="{$pwdarr[pwd_id].Email}" /></td>
    </tr>
    <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left">QQ 号：</td>
      <td height="25" align="left" valign="middle" id="right"><input id="qq" name="qq" type="text"
value="{$pwdarr[pwd_id].QQ}" /></td>
    </tr>
    <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left">邮编：</td>
      <td height="25" align="left" valign="middle" id="right"><input id="code" name="code" type="text"
value="{$pwdarr[pwd_id].code}" /></td>
    </tr>
    <tr>
      <td width="25%" height="25" align="right" valign="middle" id="left">地址：</td>
      <td height="25" align="left" valign="middle" id="right"><input id="address" name="address" type="text"
value="{$pwdarr[pwd_id].address}" /> <font color="red">*</font></td>
    </tr>
    <tr>
      <td height="30" colspan="2" align="center" valign="middle"><input name="enter" type="submit" id="enter"
value="修改" />    <input name="reset" type="reset" id="reset" value="重置" /></td>
    </tr>
</form>
</table>
{/section}
{/if}
```

3．创建脚本文件

该模块的脚本文件和用户注册模块类似，都是对信息的合法性进行验证，如信息是否为空、是否符合规范等，这里不再赘述。

4．创建处理页

当信息验证通过后，系统将跳转到处理页进行信息处理。本模块处理页分信息修改和密码修改两个页面。首先介绍信息修改页，代码如下：

代码位置：光盘\MR\22\modify_info_chk.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
require("system/system.inc.php");                                  //包含配置文件
$sql = "update tb_user set realname='".$_POST['realname']."',card='".$_POST['card']."',tel='".$_POST ['tel']."',phone='".
$_POST['phone']."',Email='".$_POST['email']."',QQ='".$_POST['qq']."',code='".$_POST['code']."',address='".$_P
OST['address']."' where id = '".$_POST['userid']."'";
$arr = $admindb->ExecSQL($sql,$conn);
if($arr)
	echo "<script>alert('修改成功');location=('index.php');</script>";
else
	echo "<script>alert('修改失败');history.go(-1);</script>";
?>
```

密码修改页的操作流程也十分类似，只是更新的数组要小得多，只有一个字段。修改密码页的代码如下：

代码位置：光盘\MR\22\modify_pwd_chk.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
require("system/system.inc.php");                                  //包含配置文件
$sql="select * from tb_user where name = '".$_SESSION['member']."' and password='".md5($_POST['old'])."' ";
$arr = $admindb->ExecSQL($sql,$conn);                              //判断用户名和密码是否正确
if($arr){
	$sql = "update tb_user set password='".md5($_POST['new1'])."' where name = '".$_SESSION['member']."'
and password='".md5($_POST['old'])."' ";                            //更新密码
	$arr = $admindb->ExecSQL($sql,$conn);
	echo "<script>alert('密码修改成功！ '); window.location.href='index.php';</script>";
}else{
	echo "<script>alert('密码修改失败！ '); window.location.href='index.php';</script>";
}
?>
```

22.8.4 安全退出

当用户需要离开网站时，可以单击“安全退出”超链接来调用 logout()函数，当用户确认退出后，则跳转到 logout 页面，销毁 session 并回到首页。安全退出所涉及的页面及代码如下：

代码位置：光盘\MR\22\js\info.js

```
function logout(){
	if(confirm("确定要退出登录吗？ ")){                 //输出选择框，用户可以单击“确认”或“取消”按钮
		window.open('logout.php','_parent','',false);     //如果用户确认退出，则打开 logout.php 页
	}else
	 return false;
}
```



代码位置：光盘\MR\22\logout.php

```
<?php
	session_start();
	header ( "Content-type: text/html; charset=UTF-8" );				//设置文件编码格式
	session_destroy();
	echo '<script>alert(\'用户已安全退出!\');location=(\'index.php\');</script>';
?>
```

22.9　商品展示模块设计

22.9.1　商品展示模块概述

本系统为用户提供了不同的商品展示方式，包括推荐商品、最新商品、热门商品等，能够使消费者有目的地选购商品。每个展示方式中包括商品的详细信息显示，为用户购买商品提供可靠的依据。本系统商品显示模块的运行结果如图 22.16 所示。



图 22.16　商品展示模块页面

因为推荐商品、最新商品和热门商品的实现方法和过程基本相同，所以本节只讲解推荐商品模块。其他功能相关代码可参见光盘中的源程序。

22.9.2　商品展示模块技术分析

商品显示功能实现的关键就是如何从数据库中读取商品信息，如何完成数据的分页显示。在定义 SQL 语句时，首先判断字段 isnom 的值，如果该字段为 1，即为推荐，否则为不推荐，然后再定义数据降幂排列，并设置每页显示 4 条记录，这就是完成商品显示的查询语句，其代码如下：

代码位置：光盘\MR\22\newhot.php

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );						//设置文件编码格式
	include_once("system/system.inc.php");							//包含类的实例化文件
	$newsql = "select id,name,pics,m_price,v_price from tb_commo where isnew = 1 order by id desc limit 4";
	//定义 SQL 语句
```

```
        $hotsql = "select id,name,pics,m_price,v_price from tb_commo order by sell,id desc limit 4";
        $sql = "select id,name,pics,m_price,v_price from tb_commo where isnom = 1 order by id desc limit 4";

        $newarr = $admindb->ExecSQL($newsql,$conn);              //执行 SQL 语句，降幂排列，显示 4 条记录
        $hotarr = $admindb->ExecSQL($hotsql,$conn);
        $nomarr = $admindb->ExecSQL($sql,$conn);
        $smarty->assign('newarr',$newarr);                        //将查询结果赋给指定的模板变量
        $smarty->assign('hotarr',$hotarr);
        $smarty->assign('nomarr',$nomarr);
?>
```

最后，定义模板文件，通过 section 语句循环输出存储在模板变量中的数据，即完成商品展示的操作。

section 是 Smarty 模板中的一个循环语句，该语句用于复杂数组的输出。其语法如下：

```
{section name="sec_name" loop=$arr_name start=num step=num}
```

其中，参数 name 是该循环的名称；loop 为循环的数组；start 表示循环的初始位置，如 start=2，那么说明循环是从 loop 数组的第二个元素开始的；step 表示步长，如 step=2，那么循环一次后，数组的指针将向下移动两位，依此类推。

22.9.3 商品展示模块的实现过程

在 22.9.2 小节中已经对商品显示所使用的技术、方法进行概述，下面就介绍它的具体过程。

（1）创建 newhot.php 文件，从数据库中读取出推荐商品的数据，并将数据存储到模板变量中，其代码可以参考 22.9.2 小节。

（2）创建 newhot.tpl 模板页，应用 section 语句输出商品信息，并添加相应的操作按钮或链接。模板页中一共有 3 个事件，即显示更多商品、查看商品和放入购物车。

☑ 当单击“更多商品”超链接时，将会重新加载本页面，并传递一个 page 变量。switch 语句会根据 page 值来显示。

☑ 当单击“查看详情”按钮时，将触发 onClick 事件，并调用 openshowcommo()函数，同时，商品 id 会作为函数的唯一参数被传递进去。

☑ 当单击“购买”按钮时，同样会触发 onClick 事件，并调用 buycommo()函数，唯一的参数也是商品的 id。

商品模板页面的代码如下：

代码位置：光盘\MR\22\system\templates\newhot.tpl

```
<link rel="stylesheet" href="css/newhot.css" />
<link href="css/top.css" rel="stylesheet" type="text/css" />
<link href="css/nominate.css" rel="stylesheet" type="text/css" />
<link href="css/links.css" rel="stylesheet" type="text/css" />
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/showcommo.js"></script>
<table width="643" border="0" cellpadding="0" cellspacing="0" style=" border: 3px solid #f0f0f0;" >
    <tr>
        <td width="321" height="33" align="center" background="images/shop_07.gif"><div class="new"><a href="?page=nom" class="top"><img src="images/more.JPG" width="39" height="18" border="0" /></a></div>
      </td>
        <td width="322" height="33" align="right" background="images/shop_14.gif"><div class="hot"><a href="?page=hot" class="top"><img src="images/more.JPG" width="39" height="18" border="0" /></a></div>
```

```
            </td>
  </tr>
     <tr>
        <td align="center" valign="top" style="border-right: 1px solid #f0f0f0;"><table width="295" height="307"
align="center" border="0" cellpadding="0" cellspacing="0">
          <tr>{counter start=1 skip=1 direction=up print=false assign=count} {section name=new_id loop=
$newarr}
            <td align="left" valign="top"><table width="150" height="150" align="left" border="0" cellpadding="0"
cellspacing="0">
                   <tr>
                     <td height="100" align="center" valign="middle"><a style="cursor:hand;" onclick=""><img
src="{$newarr[new_id].pics}" width="100" height="80" alt="{$newarr[new_id].name}" style="border:1px solid
#f0f0f0;" onclick="openshowcommo({$newarr[new_id].id})" /></a></td>
                   </tr>
                   <tr>
                     <td height="17" align="center" valign="middle">{$newarr[new_id].name}</td>
                   </tr>
                   <tr>
                     <td height="17" align="center" valign="middle">市场价：{$newarr[new_id].m_price} 
元</td>
                   </tr>
                   <tr>
                     <td height="16" align="center" valign="middle">会员价：{$newarr[new_id].v_price} 
元</td>
                   </tr>
              </table></td>
                {counter}
            {if $count mod 2 != 0} </tr>
          <tr> {/if}
             {/section} </tr>
       </table></td>
           <td align="center" valign="top" style="border-left: 1px solid #f0f0f0;"><table width="295" height="307"
align="center" border="0" cellpadding="0" cellspacing="0">
            <tr> {counter start=1 skip=1 direction=up print=false assign=counts}{section name=hot_id loop=
$hotarr}
              <td align="left" valign="top"><table width="150" height="150" align="left" border="0" cellpadding=
"0" cellspacing="0">
                     <tr>
                       <td height="100" align="center" valign="middle"><a style="cursor:hand;" onclick=""><img
src="{$hotarr[hot_id].pics}" width="100" height="80" alt="{$hotarr[hot_id].name}" style="border:1px solid
#f0f0f0;" onclick="openshowcommo({$hotarr[hot_id].id})" /></a></td>
                     </tr>
                     <tr>
                       <td height="17" align="center" valign="middle">{$hotarr[hot_id].name}</td>
                     </tr>
                     <tr>
                       <td height="17" align="center" valign="middle">市场价：{$hotarr[hot_id].m_price}</td>
                     </tr>
                     <tr>
                       <td height="16" align="center" valign="middle">会员价：{$hotarr[hot_id].v_price}</td>
                     </tr>
```

```
                  </table></td>
              {counter}
             {if $counts mod 2 != 0}</tr>
               <tr> {/if}
                 {/section} </tr>
          </table></td>
</tr>
</table>
<table width="643" border="0" cellpadding="0" cellspacing="0">
      <tr>
            <td colspan="6" width="636" height="33" align="right" valign="middle"><img src="images/shop_ 10.gif"
width ="643" height="33" border="0" usemap="#Map" /></td>
            <td rowspan="3" width="7" height="238"> </td>
      </tr>
      <tr>
      <td width="23" height="185"> </td>
         {section name=nom_id loop=$nomarr}
         <td width="145" height="185" align="left" valign="top">
               <table width="145" border="0" cellpadding="0" cellspacing="0" >
<tr>
                           <td height="100" align="center" valign="middle"><img src="{$nomarr[nom_id].pics}"
width="100" height="80" alt="{$nomarr[nom_id].name}" style="border: 1px solid #f0f0f0;" ></td>
                </tr>
                     <tr>
                           <td height="17" align="center" valign="middle"> {$nomarr[nom_id].name}</td>
                </tr>
                     <tr>
                        <td height="17" align="center" valign="middle">市场价：{$nomarr[nom_id].m_price}  元</td>
                </tr>
                     <tr>
                        <td height="19" align="center" valign="middle">会员价：{$nomarr[nom_id].v_price} 元
</td>
                </tr>
                     <tr>
                        <td height="32" align="center" valign="middle"><input id="showinfo" name="showinfo"
type="button" value="" class="showinfo" onclick="openshowcommo({$nomarr[nom_id].id})"/> <input id=
"buy" name="buy" type="button" value="" class="buy" onclick="return buycommo({$nomarr[nom_id].id})" /></td>
                </tr>
              </table>
         </td>
        {/section}
          <td width="33" height="185"> </td>
   </tr>
      <tr>
            <td colspan="6" width="636" height="14"> </td>
      </tr>
</table>
<map name="Map" id="Map">
<area shape="rect" coords="585,8,635,27" href="?page=new" class="lk" />
</map>
```

（3）创建 showcommo.js 脚本文件。当单击“查看商品”按钮时，系统会弹出一个新的页面，并显示

商品的详细信息；当单击“购买”按钮时，该商品将会被放到当前用户的购物车中，如果没有登录用户或商品已添加，则会提示错误信息。js 脚本文件的代码如下：

代码位置：光盘\MR\22\js\showcommo.js

```
/*	查看商品信息函数，将打开一个新页面	*/
function openshowcommo(key){
	open('showcommo.php?id='+key,'_blank','width=560 height=300',false);
}
/*	将购买商品添加到购物车中，将在 22.10 节中讲解	*/
function buycommo(key){
	…
}
```

22.10　购物车模块设计

22.10.1　购物车模块概述

购物车在电子商务平台中是前台客户端程序中非常关键的一个功能模块。购物车的主要功能是保留用户选择的商品信息，用户可以在购物车内设置选购商品的数量，显示选购商品的总金额，还可以清除选择的全部商品信息，重新选择商品信息。购物车页面运行结果如图 22.17 所示。

我的购物车						
	商品名称	购买数量	市场价格	会员价格	折扣率	合计
□	自行车	3	388	349.2	9	1047.6
□	数码相机	5	1888	1699.2	9	8496
□	洗衣机	1	2666	2399.4	9	2399.4
□	家庭影院	1	4888	4399.2	9	4399.2
全选 反选	删除选择		继续购物	去收银台		共计：16342.2 元

图 22.17　购物车页面

购物车模块主要实现添加商品、删除商品和更改数量等操作。

22.10.2　购物车模块技术分析

购物车功能实现最关键的部分就是如何将商品添加到购物车中，如果不能完成商品的添加，那么购物车中的其他操作都没有任何意义。

（1）在商品显示模块中，单击商品中的“购买”按钮，将商品放到购物车中，并进入到购物车页面。单击“购买”按钮调用 buycommo()函数，购买商品的 id 是该函数的唯一参数，在 buycommo()函数中通过 xmlhttp 对象调用 chklogin.php 文件，并根据回传值作出相应处理。buycommo()函数的代码如下：

代码位置：光盘\MR\22\js\showcommo.js

```
/*
*添加商品，同时检查用户是否登录、商品是否重复等
*/
function buycommo(key){
	/*	根据商品 ID，生成 url	*/
	var url = "chklogin.php?key="+key;
```

```
	/*	使用 xmlhttp 对象调用 chklogin.php 页	*/
	xmlhttp.open("GET",url,true);
	xmlhttp.onreadystatechange = function(){
		if(xmlhttp.readyState == 4){
				var msg = xmlhttp.responseText;
				/*	用户没有登录	*/
				if(msg == '2'){
					alert('请您先登录');
					return false;
				}else if(msg == '3'){
				/*	商品已添加	*/
					alert('该商品已添加');
					return false;
				}else{
				/*	显示购物车	*/
					location='index.php?page=shopcar';
				}
			}
	}
	xmlhttp.send(null);
```

（2）在 chklogin.php 文件中将商品添加到购物车中。chklogin.php 页的代码如下：

代码位置：光盘\MR\22\chklogin.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );			//设置文件编码格式
require("system/system.inc.php");					//包含配置文件
/**
	*	1 表示添加成功
	*	2 表示用户没有登录
	*	3 表示商品已添加过
	*	4 表示添加时出现错误
	*	5 表示没有商品添加
*/
$reback = '0';
if(empty($_SESSION['member'])){
	$reback = '2';
}else{
	$key = $_GET['key'];
	if($key == ''){
		$reback = '5';
	}else{
		$boo = false;
		$sqls = "select id,shopping from tb_user where name = '".$_SESSION['member']."'";
		$shopcont = $admindb->ExecSQL($sqls,$conn);
		if(!empty($shopcont[0]['shopping'])){
			$arr = explode('@',$shopcont[0]['shopping']);
			foreach($arr as $value){
				$arrtmp = explode(',',$value);
				if($key == $arrtmp[0]){
					$reback = '3';
```

```
					$boo = true;
					break;
				}
			}
			if($boo == false){
				$shopcont[0]['shopping'] .= '@'.$key.',1';
				$update = "update tb_user set shopping='".$shopcont[0]['shopping']."' where name = '".$_
SESSION ['member']."'";
				$shop = $admindb->ExecSQL($update,$conn);
				if($shop){
					$reback = 1;
				}else{
					$reback = '4';
				}
			}
		}else{
			$tmparr = $key.",1";
			$updates = "update tb_user set shopping='".$tmparr."' where name = '".$_SESSION['member']."'";
			$result = $admindb->ExecSQL($updates,$conn);
			if($result){
				$reback = 1;
			}else{
				$reback = '4';
			}
		}
	}
}
echo $reback;
?>
```

通过分析上述代码可知，shopping 字段保存的是购物车中的商品信息，一条商品信息包括两部分，即商品 id 和商品数量，其中商品数量默认为 1。两部分之间使用逗号“,”分隔，如果添加多个商品，则每个商品之间使用“@”分隔。

成功完成商品的添加操作后，即可进入到购物车页面，执行其他的操作。

22.10.3　购物车展示

购物车页面分 PHP 代码页和 Smarty 模板页。在 PHP 代码页中，首先读取 tb_user 数据表中 shopping 字段的内容，如果字段为空，则输出“暂无商品”；如果数据库中有数据，则循环输出数据，并将商品信息保存到数组中，再传给模板页。购物车页面的代码如下：

代码位置：光盘\MR\22\myshopcar.php

```
<?php
$select = "select id,shopping from tb_user where name ='".$_SESSION['member']."'";
$rst = $admindb->ExecSQL($select,$conn);
if($rst[0]['shopping']==""){
	echo "<p>";
	echo '购物车中暂时没有商品!';
	exit();
}
```

```
$commarr = array();
foreach($rst[0] as  $value){
	$tmpnum = explode('@',$value);
	$shopnum = count($tmpnum);                          //商品类数
	$sum = 0;
	foreach($tmpnum as $key => $vl){
		$s_commo = explode(',',$vl);
		$sql2 = "select id,name,m_price,fold,v_price from tb_commo";
		$commsql = $sql2." where id = ".$s_commo[0];
		$arr = $admindb->ExecSQL($commsql,$conn);
		@$arr[0]['num'] = $s_commo[1];
		@$arr[0]['total'] = $s_commo[1]*$arr[0]['v_price'];
		$sum += $arr[0]['total'];
		$commarr[$key] = $arr[0];
	}

	}
	$smarty->assign('shoparr',$shopnum);
	$smarty->assign('commarr',$commarr);
	$smarty->assign('sum',$sum);
?>
```

商品的模板页不仅要负责用户购买商品信息的输出，而且还要提供可以对商品进行修改、删除等操作的事件接口。模板页的代码如下：

代码位置：光盘\MR\22\system\templates\myshopcar.tpl

```
<table border="0" cellspacing="0" cellpadding="0" align="center">
<form id="myshopcar" name="myshopcar" method="post" action="#">
  <tr>
    <td height="30" colspan="7" align="center" valign="middle" class="first">我的购物车</td>
  </tr>
  <tr>
    <td width="35" height="25" align="center" valign="middle" class="left"> </td>
    <td width="100" height="25" align="center" valign="middle" class="center">商品名称</td>
    <td width="100" height="25" align="center" valign="middle" class="center">购买数量</td>
    <td width="100" height="25" align="center" valign="middle" class="center">市场价格</td>
    <td width="100" height="25" align="center" valign="middle" class="center">会员价格</td>
    <td width="100" height="25" align="center" valign="middle" class="center">折扣率</td>
    <td width="100" height="25" align="center" valign="middle" class="right">合计</td>
  </tr>
{foreach key=key item=item from=$commarr}
  <tr>
    <td height="25" align="center" valign="middle" class="left"><input id="chk" name="chk[]" type="checkbox"
value="{$item.id}"></td>
    <td height="25" align="center" valign="middle" class="center"><div id = "c_name{$key}">  {$item.name}
</div></td>
    <td height="25" align="center" valign="middle" class="center"><input id="cnum{$key}" name="cnum{$key}"
type="text" class="shorttxt" value="{$item.num}" onkeyup="cvp({$key},{$item.v_price},{$shoparr})"></td>
    <td height="25" align="center" valign="middle" class="center"><div id="m_price{$key}"> {$item.m_
price}</div></td>
    <td height="25" align="center" valign="middle" class="center"><div id="v_price{$key}"> {$item.v_
price}</div></td>
```

```
		<td height="25" align="center" valign="middle" class="center"><div id="fold{$key}"> {$item.fold}</div></td>
		<td height="25" align="center" valign="middle" class="right"><div id="total{$key}"> {$item.total}</div></td>
	</tr>
{/foreach}
	<tr>
		<td height="25" colspan="3" align="left" valign="middle">
		<a href="#" onclick="return alldel(myshopcar)">全选</a> <a href="#" onclick="return overdel(myshopcar);">反选</a>  
			<input type="button" value="删除选择" class="btn" style="border-color: #FFFFFF;" onClick = 'return del(myshopcar);'>
  	</td>
		<td height="25" align="center" valign="middle"><input id="cont" name="cont" type="button" class="btn" value="继续购物" onclick="return conshop(myshopcar)" /></td>
		<td height="25" align="center" valign="middle"><input id="uid" name="uid" type="hidden" value="{$smarty.session.member}" ><input id="settle" name="settle" type="button" class="btn" value="去收银台" onclick="return formset(form)" /></td>
		<td height="25" colspan="2" align="right" valign="middle"><div id='sum'>共计：{$sum} 元</div></td>
		</tr>
</form>
</table>
```

22.10.4 更改商品数量

对于新添加的商品，默认的购买数量为 1，在购物车页面可以对商品的数量进行修改。当商品数量发生变化时商品的“合计”金额和商品总金额会自动发生改变，该功能是通过触发 text 文本域的 onKeyUp 事件调用 cvp()函数实现。cvp()函数有 3 个参数，分别是商品 id、商品单价和商品类别。

首先，通过商品的 id 可以得到要修改商品的相关表单和标签属性。然后，通过商品单价和输入的商品数量计算该商品的合计金额。接着，使用 for 循环得到其他商品的合计金额。最后，将所有的合计金额累加，并输出到购物车页面。cvp()函数的代码如下：

代码位置：光盘\MR\22\js\shopcar.js

```
function cvp(key,vpr,shoparr){
	var n_pre = 'total';
	var num = 'cnum'+key.toString();
	var total = n_pre+key.toString();
	var t_number = document.getElementById(num).value;
	var ttl = t_number * vpr;
	document.getElementById(total).innerHTML = ttl;
	var sm = 0;

	for(var i = 0; i < shoparr; i++){

		var aaa = document.getElementById(n_pre+i.toString()).innerText;
		sm += parseInt(aaa);
	}
	document.getElementById('sum').innerHTML = '共计：'+sm+' 元';

}
```

这里所更改的商品数量，并没有被保存到数据库中，如果希望保存，那么单击“继续购物”按钮，则

可以将商品数量更新到数据库中。

22.10.5　删除商品

当对添加的商品不满意时，可以对商品进行删除操作。操作流程为：首先选中要删除的商品前面的复选框，如果全部删除，则可以单击“全选”按钮或“反选”按钮；然后单击“删除选择”按钮，在弹出的警告框中单击“确定”按钮，商品将被全部删除。删除商品的页面结果如图 22.18 所示。

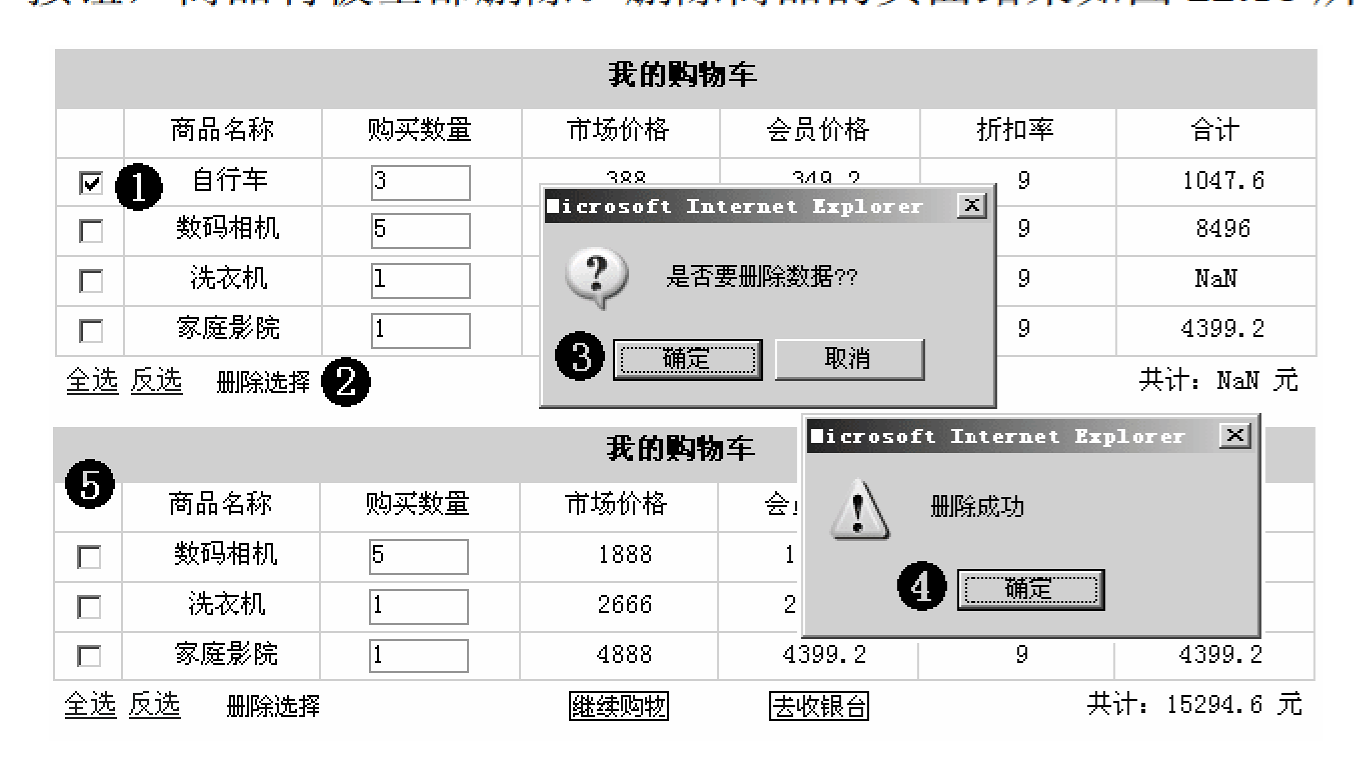


图 22.18　删除商品流程

所有的删除操作都是通过 js 脚本文件 shopcar.js 来实现的，相关的函数包括 alldel()函数、overdel()函数和 del()函数。

alldel()函数和 overdel()函数实现的原理比较简单，通过触发 onClick 事件来改变复选框的选中状态。这两个函数的代码如下：

代码位置：光盘\MR\22\js\shopcar.js

```
//全部选择/取消
function alldel(form){
	var leng = form.chk.length;
	if(leng==undefined){
	    if(!form.chk.checked)
		    form.chk.checked=true;
	 }else{
	   for( var i = 0; i < leng; i++)
	     {
		    if(!form.chk[i].checked)
			    form.chk[i].checked = true;
	     }
	 }
	return false;
}
//反选
function overdel(form){
	 var leng = form.chk.length;
	 if(leng==undefined){
	    if(!form.chk.checked)
		    form.chk.checked=true;
```

```
		else
			form.chk.checked=false;
	}else{
	  for( var i = 0; i < leng; i++)
	    {
			if(!form.chk[i].checked)
				form.chk[i].checked = true;
			else
				form.chk[i].checked = false;
	    }
	}
	return false;
}
```

使用 alldel()或 overdel()选中复选框后，即可调用 del()函数来实现删除功能。del()函数首先使用 for 循环，将被选中复选框的 value 值取出并存成数组，然后根据数组生成 url，并使用 xmlhttp 对象调用这个 url，当处理完毕后，根据返回值弹出提示或刷新本页。该函数的代码如下：

代码位置：光盘\MR\22\js\shopcar.js

```
/*   删除记录   */
function del(form){
	if(!window.confirm('是否要删除数据??')){

	}else{
		var leng = form.chk.length;
		if(leng==undefined){
			if(!form.chk.checked){
					alert('请选取要删除的数据!');
			}else{
				rd = form.chk.value;
				var url = 'delshop.php?rd='+rd;
				xmlhttp.open("GET",url,true);
				xmlhttp.onreadystatechange = delnow;
				xmlhttp.send(null);
			}
		}else{
			var rd=new Array();
			var j = 0;
			for( var i = 0; i < leng; i++)
			{
				if(form.chk[i].checked){
					rd[j++] = form.chk[i].value;
				}
			}
			if(rd == ''){
				alert('请选取要删除的数据!');
			}else{
				var url = "delshop.php?rd="+rd;
				xmlhttp.open("GET",url,true);
				xmlhttp.onreadystatechange = delnow;
				xmlhttp.send(null);
			}
```

```
			}
		}
		return false;
}
function delnow(){
		if(xmlhttp.readyState == 4){
			if(xmlhttp.status == 200){
				var msg = xmlhttp.responseText;
				if(msg != '1'){
					alert('删除失败'+msg);
				}else{
					alert('删除成功');
					location=('?page=shopcar');
				}
			}
		}
}
```

22.10.6 保存购物车

当用户希望保存商品更改后的商品数量时，可以单击“继续购物”按钮，将触发 onClick 事件并调用 conshop()函数保存数据，该函数有一个参数，就是当前表单的名称。在 conshop()函数内，根据复选框和商品数量文本域，生成两个数组 fst 和 snd，分别保存商品 id 和商品数量。

这里要注意，两个数组的值是要相互对应的，如商品 1 的 id 保存到 fst[1]中，那么商品 1 的数量就要保存到 snd[1]中，然后根据这两个数组生成一个 url，使用 xmlhttprequest 对象调用 url，最后根据回传信息作出相应的判断。conshop()函数的代码如下：

代码位置：光盘\MR\22\js\shopcar.js

```
//更改商品数量
function conshop(form){
		var n_pre = 'cnum';
		var lang = form.chk.length;
		if(lang == undefined){
			var fst = form.chk.value;
			var snd = form.cnum0.value;
		}else{
			var fst= new Array();
			var snd = new Array();
			for(var i = 0; i < lang; i++){
				var nm = n_pre+i.toString();
				var stmp = document.getElementById(nm).value;
				if(stmp   == '' || isNaN(stmp)){
					alert('不允许为空、必须为数字');
					document.getElementById(nm).select();
					return false;
				}
				snd[i] = stmp;
				var ftmp = form.chk[i].value;
				fst[i] = ftmp;
```

```
		}
	}
	var url = 'changecar.php?fst='+fst+'&snd='+snd;
	xmlhttp.open("GET",url,true);
	xmlhttp.onreadystatechange = updatecar;
	xmlhttp.send(null);
}
function updatecar(){
	if(xmlhttp.readyState == 4){
		var msg = xmlhttp.responseText;
			if(msg == '1'){
				location='index.php';
			}else{
				alert('操作失败'+msg);
			}
	}
}
```

在 conshop()函数中调用的 changecar.php 页为数据处理页，该页将商品 id 和商品数量进行重新排列，并保存到 shopping 字段内。changecar.php 页面的代码如下：

代码位置：光盘\MR\22\changecar.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );			//设置文件编码格式
require("system/system.inc.php");						//包含配置文件
$sql = "select id,shopping from tb_user where name = '".$_SESSION['member']."'";
$rst = $admindb->ExecSQL($sql,$conn);
$reback = '0';
$changecar = array();
if(isset($_GET['fst']) && isset($_GET['snd'])){
	$fst = $_GET['fst'];
	$snd = $_GET['snd'];
	$farr = explode(',',$fst);
	$sarr = explode(',',$snd);
	$upcar = array();
	for($i = 0; $i < count($farr); $i++){
		$upcar[$i] = $farr[$i].','.$sarr[$i];
	}
	if(count($farr) > 1){
		$update = "update tb_user set shopping='".implode('@',$upcar)."' where name = '".$_SESSION['member']."'";
	}else{
		$update = "update tb_user set shopping='".$upcar[0]."' where name = '".$_SESSION['member']."'";
	}
	$shop = $admindb->ExecSQL($update,$conn);
	if($shop){
		$reback = 1;
	}else{
		$reback = 2;
	}
}
```

```
echo $reback;
?>
```

22.11　收银台模块设计

22.11.1　收银台模块概述

当用户停止浏览商品准备结账时，可以单击购物车页面中的"去收银台"按钮，该按钮将触发 onClick 事件并调用 formset()函数显示订单页面，当用户提交订单后，系统将订单保存到数据表 tb_form 中，同时清空购物车，并显示订单信息提醒用户记录订单号。当货款发出后，还可以对订单进行查询。收银台页面的运行结果如图 22.19 所示。

本节所涉及的页面有显示订单（formset()函数）、填写订单（settle.php、settle.tpl）、处理订单（settle_chk.php）、反馈订单（forminfo.php、forminfo.tpl）和查询订单 5 部分。

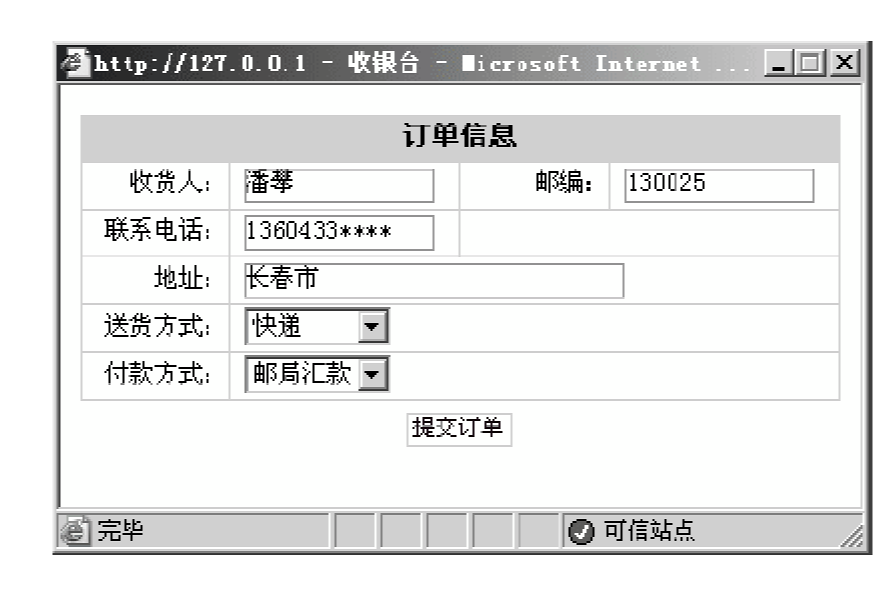


图 22.19　收银台页面运行结果

22.11.2　收银台模块技术分析

在收银台模块中，通过 PDO 中的方法完成订单信息的添加和数据的更新操作。同样调用数据库管理类 AdminDB 中的 ExecSQL()方法，完成数据的添加操作。

22.11.3　显示订单

订单信息提交页面的输出由 formset()函数决定，它对商品信息进行整理，通过 open 方法打开 settle.php 页来显示订单，并将整理后的商品信息传递到 settle.php 文件中。formset()函数的代码如下：

代码位置：光盘\MR\22\js\shopcar.js

```
function formset(form){
var uid = form.uid.value;
var n_pre = 'cnum';                                   //数量
    var lang = form.chk.length;
    if(lang == undefined){
        var fst = form.chk.value;                     //商品 id
        var snd = form.cnum0.value;                   //购买数量
    }else{
        var fst= new Array();
        var snd = new Array();
        for(var i = 0; i < lang; i++){
            var nm = n_pre+i.toString();
            var stmp = document.getElementById(nm).value;
            if(stmp   == '' || isNaN(stmp)){
                alert('不允许为空、必须为数字');
                document.getElementById(nm).select();
                return false;
```

```
                    }
                    snd[i] = stmp;
                    var ftmp = form.chk[i].value;
                    fst[i] = ftmp;
            }
        }
        open('settle.php?uid='+uid+'&fst='+fst+'&snd='+snd,'_blank','width=500 height=450',false);
}
```

说明：因为 open 方法使用了_blank 参数来打开一个新的页面，session 值传不过去，所以这里使用隐藏域来传递用户名称。

22.11.4　填写订单

settle.php 直接将接收的值传给 settle.tpl 模板，并载入 settle.tpl 模板。settle.php 页面的代码如下：

代码位置：光盘\MR\22\settle.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
require("system/system.inc.php");                                  //包含配置文件
$fst = $_GET['fst'];
$snd = $_GET['snd'];
$uid = $_GET['uid'];
$smarty->assign('title','收银台');
$smarty->assign('fst',$fst);
$smarty->assign('snd',$snd);
$smarty->assign('uid',$uid);
$smarty->display('settle.tpl');
?>
```

settle.tpl 模板显示一个表单，这个表单的内容需要用户来填写，包括收货人、联系电话等信息。而从 PHP 页传过来的几个变量则被保存到隐藏域以传递到处理页，在表单中将数据提交到 settle_chk.php 处理页。

22.11.5　处理订单

处理页 settle_chk.php 获取表单中提交的数据，根据用户提交的商品信息重新查找数据表 tb_commo，并从数据表中提取商品信息，保存到数组中，然后处理页将数组作为一条记录添加到表 tb_form 内。

数据添加成功的同时，处理页会根据 uid 找到该用户，将 shopping 字段清空，最后调用 forminfo.php 页来显示新添加的订单信息。settle_chk.php 页的代码如下：

代码位置：光盘\MR\22\settle_chk.php

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
require("system/system.inc.php");                                  //包含配置文件
$sql="insert into tb_form(formid,commo_id,commo_name,commo_num,agoprice,fold,total, vendee,taker,address, tel,
code,pay_method,del_method,formtime,state)values(";
$formid=time();
$tmpid = explode(',',$_POST['fst']);
$tmpnm = explode(',',$_POST['snd']);
```

```
$number = count($tmpid);
$tmpna = array();
$tmpvp = array();
$tmpfd = array();
$tmptt = 0;
if($number >1){
	for($i = 0; $i < $number; $i++){
		$tmpsql = "select name,v_price,fold from tb_commo where id = '".$tmpid[$i]."'";
		$tmprst = $admindb->ExecSQL($tmpsql,$conn);
		$tmpna[$i] = $tmprst[0]['name'];
		$tmpvp[$i] = $tmprst[0]['v_price'];
		$tmpfd[$i] = $tmprst[0]['fold'];
		$tmptt += $tmprst[0]['v_price'] * $tmpnm[$i];
		@$tmpsell = $tmprst[0]['sell'] + 1;
		$addsql = "update tb_commo set sell = '".$tmpsell."' where id = '".$tmpid[$i]."'";
		$addrst = $admindb->ExecSQL($addsql,$conn);
	}

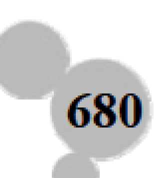
	$sql.="'".$formid."','".$_POST['fst']."','".implode(',',$tmpna)."','".$_POST['snd']."','".implode(',',$tmpvp)."',
'".implode(',',$tmpfd)."','".$tmptt."','".$_POST['uid']."'";

}else if($number == 1){
	$tmpsql = "select name,v_price,fold from tb_commo where id = '".$tmpid[0]."'";
	$tmprst = $admindb->ExecSQL($tmpsql,$conn);
	$tmptt= $tmprst[0]['v_price'] * $tmpnm[0];
	@$tmpsell = $tmprst[0]['sell'] + 1;
	$addsql = "update tb_commo set sell = '".$tmpsell."' where id = '".$tmpid[0]."'";
	$addrst = $admindb->ExecSQL($addsql,$conn);

	$sql.="'".$formid."','".$_POST['fst']."','".$tmprst[0]['name']."','".$_POST['snd']."','".$tmprst[0]['v_price']."',
'".$tmprst[0]['fold']."','".$tmptt."','".$_POST['uid']."'";
}else{
	echo 'error';
	exit();
}
$sql.=",'".$_POST['taker']."','".$_POST['address']."','".$_POST['tel']."','".$_POST['code']."','".$_POST['pay']."',
'".$_POST['del']."','".date("Y-m-d H:i:s")."',0)";
$InsertSQL = $admindb->ExecSQL($sql,$conn);
if(false == $InsertSQL){
	echo "<script>alert('购买失败');history.back;</script>";
}else{
	$updsql = "update tb_user set consume='".$tmptt."',shopping='' where name = '".$_POST['uid']."'";
	$updrst = $admindb->ExecSQL($updsql,$conn);
	echo "<script>top.opener.location.reload();</script>";
	echo                                   "<script>open('forminfo.php?fid=$formid','_blank','width=750
height=650',false);window.close();</script>";

}
?>
```

由于篇幅所限，有关反馈订单和查询订单的内容这里不再讲解，请读者参考本书光盘中的源代码。



22.12　后台首页设计

22.12.1　后台首页概述

后台管理系统是网站管理员对商品、会员及公告等信息进行统一管理的场所，本系统的后台主要包括以下几个功能。

- ☑ 类别管理模块：主要包括对商品类别的添加、修改及删除操作。
- ☑ 商品管理模块：主要包括对商品的添加、修改、删除及订单处理。
- ☑ 用户管理模块：主要包括管理员管理和会员管理。其中管理员管理是实现对管理员的添加、删除和修改功能，会员管理则包括删除和冻结功能。
- ☑ 公告管理模块：主要包括公告的添加及删除操作。
- ☑ 链接管理模块：主要包括添加、修改和删除友情链接。

后台首页的运行结果如图 22.20 所示。



图 22.20　后台首页运行结果

22.12.2　后台首页技术分析

后台首页和前台首页不同，其使用的是框架布局。框架布局的特点是，可以将容器窗口划分为若干个子窗口，每个子窗口可以分别显示不同的网页，网页之间是相互独立的，没有直接的关联，再由一个网页将这些分开的网页组成一个完整的网页，显示在浏览者的浏览器中。框架布局的好处是，每次浏览者发出对页面的请求时，只下载发生变化的框架页面，其他子页面保持不变。下面来具体看一下框架布局的使用格式及属性。

1．框架布局格式

框架布局的格式很简单，只要几行代码即可，常用的格式如下：

```
<html>
<head>
…
</head>
<frameset>
    <frame>
```

```
        <frame>
</frameset>
<noframes>
        <body>
        …
        </body>
</noframes>
</html>
```

其中<frameset>和<frame>标签是框架集标记，而<noframes>标签是为了防止浏览器不支持框架而实行的一种补救措施。如果浏览器不支持框架集，就会执行<noframes>标记中的内容，让用户能够正常浏览网页。

2. 框架集属性

框架集包含各个框架的信息，通过<frameset>标记来定义。框架是按照行和列来组织的，可以使用 FRAMESET 标记的属性对框架的结构进行设置。下面给出框架集的常用属性、说明和举例，如表 22.1 所示。

表 22.1 框架集的常用属性、说明和举例

参　数	说　明	举　例
COLS	在水平方向上将浏览器分割成多个窗口，取值有 3 种形式，即像素、百分比（%）和相对尺寸（*）	<frameset cols="25%,100,*" > <frame></frame> </frameset>
ROWS	在垂直方向上将浏览器分割成多个窗口，取值和 COLS 类似，也是 3 种形式	<frameset rows="25%,100,*" > <frame> <frame> </frameset>
FRAMEBORDER	指定框架周围是否显示边框，取值为 1（显示边框，默认值）或 0（不显示边框）	<framset cols="25%,*" cols="*" frameborder="0"> … </frameset>
FRAMESPACING	指定框架之间的间隔，以像素为单位。默认是无间隔的	<framset cols="25%,*" cols="*" framespacing="1"> … </frameset>
BORDER	指定边框的宽度，frameborder 属性为 1 时该属性才有效	<framset cols="25%,*" cols="*" frameborder="1" border="5"> … </frameset>

3. 框架属性

使用 FRAME 标记可以设置框架的属性，包括框架的名称，框架是否包含滚动条以及在框架中显示的网页等。FRAME 标记的常用属性及其说明如表 22.2 所示。

表 22.2 FRAME 标记的常用属性及其说明

参　数	说　明
NAME	指定框架的名称
SRC	指定在框架中显示的网页文件（包括 HMRL、PHP、JSP 等网页文件）
FRAMEBODER	指定框架周围是否显示边框，取值为 1（显示边框，为默认）或 0（不显示边框）
NORESIZE	可选属性，若指定了该属性，则不能调整框架的大小
SCROLLING	指定框架是否包含滚动条。属性可以是 yes（有）、no（没有）和 auto（自由）

22.12.3　后台首页实现过程

（1）定义框架页面 main.tpl，包含 top.tpl、left.php 和 default.php 3 个文件。main.tpl 页的代码如下：

代码位置：光盘\MR\22\admin\system\templates\main.tpl

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>明日购物商城后台管理系统</title>
<link rel="stylesheet" href="css/style.css" />
</head>
<frameset rows="113,*,100" cols="1004" frameborder="no" border="0" framespacing="0">
  <frame src="top.php" name="topFrame" scrolling="No" noresize="noresize" id="topFrame" title="topFrame" />
  <frameset rows="*" cols="10%,210,*,10%" framespacing="0" frameborder="no" border="0">

<frame src="s.php" name="lFrame" frameborder="0" scrolling="auto" noresize="noresize" id="lFrame" title=
"leftFrame" />

     <frame src="left.php" name="leftFrame" frameborder="0" scrolling="auto" noresize="noresize" id="leftFrame"
title= "leftFrame" />
     <frame src="default.php" name="mainFrame" id="mainFrame" title="mainFrame" />
<frame src="s.php" name="rFrame" frameborder="0" scrolling="auto" noresize="noresize" id="rFrame"
title="leftFrame" />
  </frameset>
   <frame src="bottom.php" name="bottomFrame" scrolling="No" noresize="noresize" id="bottomFrame"
title="bottomFrame" />
</frameset>
<noframes><body>
</body>
</noframes>
</html>
```

（2）left.php 页是一个树形菜单，应用 DIV+JavaScript+CSS 来实现的。首先介绍 div 标签，在 left.tpl 模板文件中，其关键代码如下：

代码位置：光盘\MR\22\admin\system\templates\left.tpl

```
<!--  载入 CSS 样式和 JavaScript 脚本  -->
<link href="css/left.css" rel="stylesheet" type="text/css" />
<script language="javascript" src="js/left.js"></script>
<!--  类别管理菜单，注意加粗的地方  -->
<div id="type" align="center" onclick="javascript:change(one,type);">类别管理</div>
<!--  子菜单  -->
<div id="one" style="display: ">
<div id="addtype" align="center"><a href="addtype.php" target="mainFrame" id="menu">添加类别</a></div>
<div id="showtype" align="center"><a href="showtype.php" target="mainFrame" id="menu">查看类别</a></div>
</div>
<div id="hidediv" align="center"></div>
<!--  商品管理菜单  -->
<div id="commo" align="center" onclick="javascript:change(two,type);">类别管理</div>
```

```
<div id="two"style="display:none">
<!--   商品管理子菜单   -->
…
</div>
…
```

说明：除了加粗的 id 名称和 JavaScript 事件不同外，其他菜单的结构完全相同，此时只需修改超链接即可。

注意：除了第一个类别菜单的子菜单 display 样式为空外，其他几个子菜单的 div 样式都为 display= none;。

该页面在 Dreamweaver 中的效果如图 22.21 所示。



图 22.21　div 树形菜单

因为其他子菜单的样式为 display=none，所以只有“类别管理”子菜单是可见的，下面为它添加 JavaScript 事件。left.js 脚本文件的代码如下：

代码位置：光盘\MR\22\admin\js\left.js

```
function change(nu,lx){
	if(nu.style.display == "none"){
		nu.style.display = "";
		lx.style.background="url(images/admin(5).gif)";
	}else{
		nu.style.display = "none";
		lx.style.background="url(images/admin(1).gif)";
	}
}
```

最后在 left.css 中设置 div 的长、宽等一些默认参数。至此，一个简单而又实用的树形菜单就完成了。

对于后台的大部分模块来说，其功能实现的方法和开发步骤在前台的模块设计中基本都已经介绍过。由于篇幅所限，这里不再对后台管理模块进行详细的讲解。

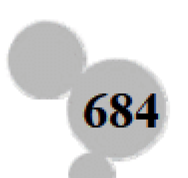
22.13　开发技巧与难点分析

在本系统开发和后期测试的过程中，开发人员遇到了各种各样的疑难问题。这里找出一些常见的、容易被忽略的问题加以讲解，希望能够为初学者和新手提供一些帮助，在开发程序时少走一些弯路。

22.13.1　解决 Ajax 的乱码问题

问题描述：当使用 Ajax 传递数据时，要么在数据处理页中数据不能被正确处理，要么输出返回值时显示的是一堆无法识别的乱码。

解决方法：这是因为 PHP 在传递数据时使用的编码默认为 UTF-8，这样就造成了非英文字符不能正确



传递的情况。解决方法如下：

在所有的 PHP 页中都输入代码“header ("Content-type: text/html; charset=UTF-8");”这样，所有的页面即可正确显示。

22.13.2　使用 JS 脚本获取、输出标签内容

问题描述：获取、更改表单元素值和特定标签内容。

解决方法：使用 JS 脚本获取页面内容的方式主要有两种，第一种是通过表单获取表单元素的 value 值。格式为表单名称.元素名.value。该方式只能获取表单中的元素值，对于其他标签元素则无能为力。而第二种方式可以通过 id 名来获取页面中任意标签的内容。格式如下：

```
document.getElementById("id"). value;或 document.getElementById ("id").innerText;
```

使用第二种方式时要注意，标签的 id 名必须存在且唯一，否则就会出现错误。为标签内容赋值时，则使用如下格式：

```
id.innerHMRL ='要显示的内容';
```

22.13.3　禁用页面缓存

问题描述：使用 Ajax 技术可以防止页面刷新，但有时也会产生新的问题。例如，在“会员管理”页面，如果连续地“冻结”和“解冻”会员，那么超过 3 次后，该功能将失效，因为在一定时间内，如果做相同的操作，那么 xmlhttprequest 对象会执行缓存中的信息，从而造成操作失败。

解决办法：使用 header()函数将缓存关闭。将代码 header("CACHE-CONTROL:NO-CACHE");添加到 xmlhttprequest 对象所调用的处理页的顶部即可。

22.13.4　在新窗口中使用 session

问题描述：使用 JavaScript 的 open 方法打开新窗口时，原浏览器中的 session 值不会被传递到新窗口中，这样会造成数据查询失败。

解决方法：将 session 值另存到隐藏域或随着 url 一起传递到新窗口。代码如下：

```
<!--   在模板页中，将 session 值赋给隐藏域   -->
<input id="uid" name="uid" type="hidden" value="{$smarty.session.id}">
…
/*     在 js 脚本中，获取到隐藏域 value 值 */
function getInput(){
      Var uid = document.getElementById('uid').value;
/*     将获取的 value 值通过 url 传给新页面       */
      open("operator.php?uid="+uid,'_blank','',false);
      …
}
```

22.13.5　判断上传文件格式

问题描述：添加商品时可以上传商品的图片，但有时可能会误传非图片格式的文件，这里就自定义一个函数来判断上传文件的后缀。

解决方法：创建自定义函数 f_postfix()，该函数的代码如下：

```
/*
 *判断文件后缀
 *$f_type：允许文件的后缀类型（数组）
 *$f_upfiles：上传文件名
 */
function f_postfix($f_type,$f_upfiles){
      $is_pass = false;
      $MRp_upfiles = split("\.",$f_upfiles);                    //使用 split()函数分隔文件
      $MRp_num = count($MRp_upfiles);                           //查找文件后缀
      if(in_array(strtolower($MRp_upfiles[$MRp_num - 1]),$f_type))  //判断后缀是否在允许列表内
            $is_pass = $MRp_upfiles[$MRp_num - 1];              //如果是，则将后缀名赋给变量
      return $is_pass;                                          //返回变量
}
```

22.13.6 设置服务器的时间

问题描述：如果没有对 PHP 的时区进行设置，那么使用日期、时间函数获取的将是英国伦敦本地时间（即零时区的时间）。例如，以东八区为例，如果当地使用的是北京时间，又没有对 PHP 的时区进行设置，那么获取的时间将比当地的北京时间少 8 个小时。

解决方案：要获取本地当前的时间必须更改 PHP 语言中的时区设置。更改 PHP 语言中的时区设置有两种方法：

（1）在 php.ini 文件中，定位到[date]下的“;date.timezone =”选项，去掉前面的分号，并设置它的值为当地所在时区使用的时间。修改内容如图 22.22 所示。

例如，如果当地所在时区为东八区，那么就可以设置“date.timezone =”的值为 PRC、Asia/Hong_Kong、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐）等。这些都是东八区的时间。

```
php.ini - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
[Date]
; Defines the default timezone used by the
date functions
; http://php.net/date.timezone
date.timezone =prc
```

图 22.22　设置 PHP 的时区

设置完成后，保存文件，重新启动 Apache 服务器。

（2）在应用程序中，在日期、时间函数之前使用 date_default_timezone_set()函数就可以完成对时区的设置。date_default_timezone_set()函数的语法如下：

```
date_default_timezone_set(timezone);
```

参数 timezone 为 PHP 可识别的时区名称，如果时区名称 PHP 无法识别，则系统采用 UTC 时区。

例如，设置北京时间可以使用的时区包括 PRC（中华人民共和国）、Asia/Chongqing（重庆）、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐），这几个时区名称是等效的。

22.14 开 发 总 结

本堂课使用 Smarty、PDO 数据库抽象层和 Ajax 等目前的主流技术，开发了一个电子商务平台。希望读者能通过该项目实例，把前面所学到的各种技术消化吸收、融会贯通，并能够学以致用、举一反三。



第23堂课

Zend Framework 框架开发明日搜索引擎

（视频讲解：80 分钟）

在全球信息化高速发展的今天，互联网已经成为资源共享、互动、交流、办公及娱乐的最为广泛和有效的平台，如何高效地利用这些资源已经成为网民们最为关注的问题。百度、Google 等互联网搜索工具的出现已经在很大程度上提高了信息共享的进程。本堂课所介绍的“明日知道”网站是针对 IT 技术问题进行搜索、为程序开发人员提供交互的平台，与互联网上一些知名的搜索工具相比更具专业化特点。

学习摘要：

- 明日搜索引擎概述
- 开发背景
- 需求分析
- 系统分析
- 数据库设计
- 技术攻关
- MVC 框架结构搭建
- 首页设计
- 用户模块设计
- 发表问题模块设计
- 回复问题模块设计
- 程序调试与错误处理

23.1　明日搜索引擎概述

在互联网高速发展的今天，网上资源积累已经相当丰富，浏览者借助互联网搜索工具（如百度、Google等）可以对网上资源进行检索，提高了现代化生活和学习的进程。“明日搜索引擎”网站是为了使搜索变得更加专业、方便广大程序相关工作者而打造的专业化互联网搜索引擎。通过该搜索引擎，程序开发人员可以方便地查找到有关软件开发方面的技术问题和文章。并且“明日搜索引擎”网站提供了互动平台，在学习编程过程中如果遇到了技术问题，可以将问题发表于明日搜索引擎网站上，其他注册会员可以对提出的问题进行解答，这样即便通过“明日搜索引擎”没有搜索到要查找的内容，也可以通过互动方式将问题解决。专业化搜索工具已经成为网上资源相当丰富的新生儿，是有一定发展潜能的项目课题。

23.2　开 发 背 景

××科技有限公司是一家以 IT 教育、互联网开发和运营为主的高科技互联网企业。在多年的经营中积累了大量的 IT 技术方面的资源，为了提升公司的知名度，使资源得以共享，并创造更大的广告利润，公司组织了一支团队来开发“明日搜索引擎”网站，争取打造一个国内知名的专业 IT 技术搜索和交互平台。例如，用户可以通过“明日搜索引擎”网站查询学习或开发项目中所遇到的技术问题，如果没有查询到，还可以将问题发表在“明日搜索引擎”网站的用户交互平台，其他用户就可以对问题进行解答。通过“明日搜索引擎”网站不仅可以解决用户学习和工作中所遇到的实际问题，而且可以进一步积累资源。

23.3　需 求 分 析

对于“明日搜索引擎”这类信息检索类网站来说，功能强大的搜索模块显得尤为重要，借鉴其他知名的搜索工具，方便、高效的搜索引擎应该具备关键字提示列表、分词查询和查询结果的描红输出等功能。为了避免用户要查找的内容在本站没有查找到结果，还需要增加用户交流模块，这样即使用户的问题在本站没有查找到答案，也可以将问题发布出去，其他用户就可以对该问题进行回答，直到解决为止。所以“明日搜索引擎”网站应该打造一个服务开发人员、对资源进行不断积累的综合平台。

通过实际调查，要求“明日搜索引擎”网站应该具备以下功能：

☑　类似百度和 Google 等搜索引擎的搜索条。
☑　多关键字的分词查询。
☑　查询结果的关键字描红。
☑　使用在线编辑器发布和回复问题。
☑　问题的引用。

23.4　系 统 分 析

“磨刀不误砍柴工”，建立一个完善的系统功能图对项目的整体开发具有指导意义，同时确立系统开发



所使用的环境也是非常必要的。

23.4.1　功能结构

在项目开发之前，设计项目前台和后台的系统功能结构图。其中“明日搜索引擎”网站前台功能结构图如图 23.1 所示。“明日搜索引擎”网站后台功能结构图如图 23.2 所示。

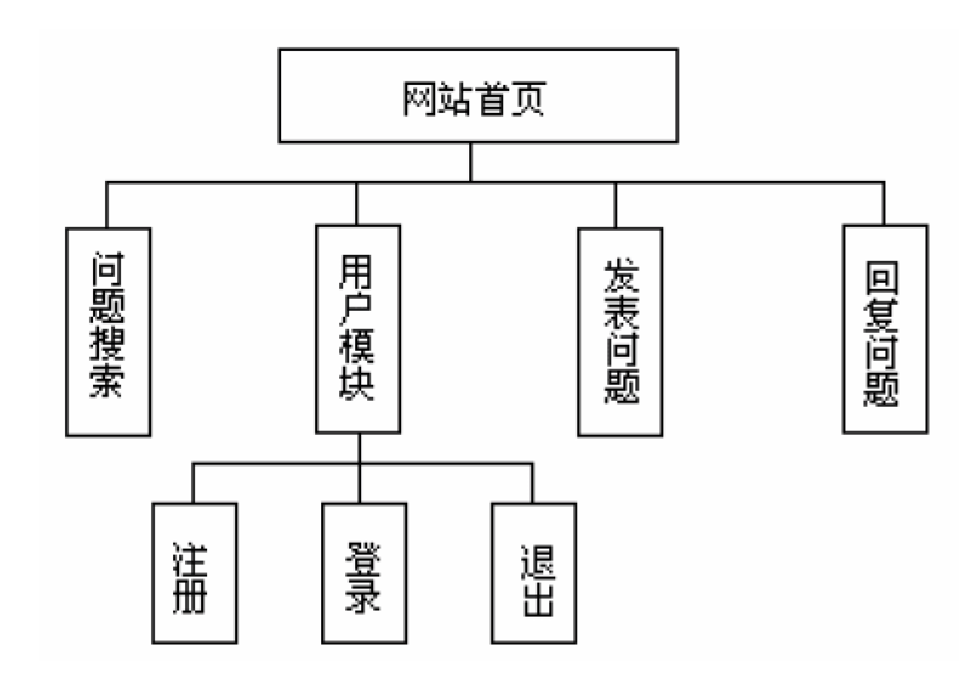


图 23.1　网站前台功能结构图

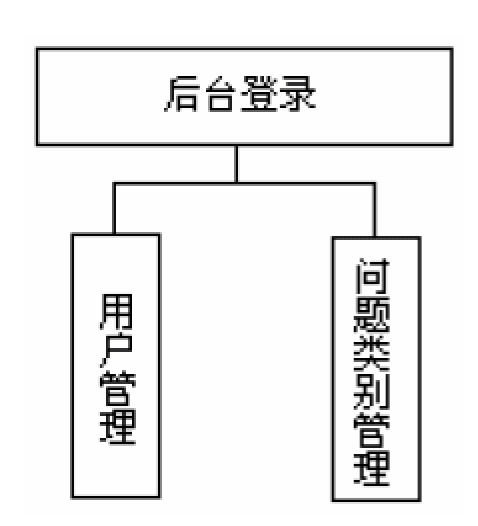


图 23.2　网站后台功能结构图

23.4.2　开发环境

在开发明日搜索引擎时，该项目使用的软件开发环境介绍如下。

1．服务器端

☑ 操作系统：Windows Server 2003/Linux（推荐）。
☑ 服务器：Apache 2.2.8。
☑ PHP 软件：PHP 5.2.6。
☑ 数据库：MySQL 5.0.51。
☑ MySQL 图形化管理软件：phpMyAdmin-2.22.3。
☑ Zend Framework 框架。
☑ 开发工具：Dreamweaver 8 或者 Zend Studio。
☑ 浏览器：IE 6.0 及以上版本。
☑ 分辨率：最佳效果为 1024 像素×768 像素。

2．客户端

☑ 浏览器：推荐 IE 6.0 及以上版本。
☑ 分辨率：最佳效果为 1024 像素×768 像素。

23.5　数据库设计

在进行项目开发前需要根据实体属性建立数据库实体模型，然后根据实体之间存在的客观联系设计出合理的数据表。由于数据库是项目内容的主要载体，所以设计出合理的数据库及表间关系显得非常重要。

23.5.1　数据库分析

“明日搜索引擎”网站采用 PHP 语言开发，并且该项目为资源搜索型网站，数据量会很大，所以数据库稳定、高效的存储和检索能力显得非常重要，综合上述因素及 PHP 的完全跨平台等特性，应该首选 MySQL 数据库作为“明日搜索引擎”网站数据的载体。在长时间的生产中也总结出，PHP 和 MySQL 是最佳的黄金搭档。

23.5.2　数据库概念设计

根据“明日搜索引擎”网站的需求分析、功能结构分析，规划出整个项目的数据库实体应该包括问题类别实体、关键字实体、问题实体、回复实体和用户实体等部分。下面介绍这几个实体的 E-R 图。

1. 问题类别实体

“明日搜索引擎”网站定位打造一个专业、高效的在线搜索工具，这样就需要搜索内容按类别划分。资源分类明确，可以有效地提高用户检索的效率，方便用户定位查找。问题类别实体包括类别 ID 号、类别名称、类别描述和类别添加时间等属性，其实体 E-R 图如图 23.3 所示。

2. 问题实体

问题实体是对用户提出问题的抽象，根据功能分析，问题实体包括问题 ID 号、主题、内容、添加时间、置顶标识、提问用户 ID 号、问题类别 ID 号、浏览次数、上传文件名和精华问题标识等属性，其实体 E-R 图如图 23.4 所示。

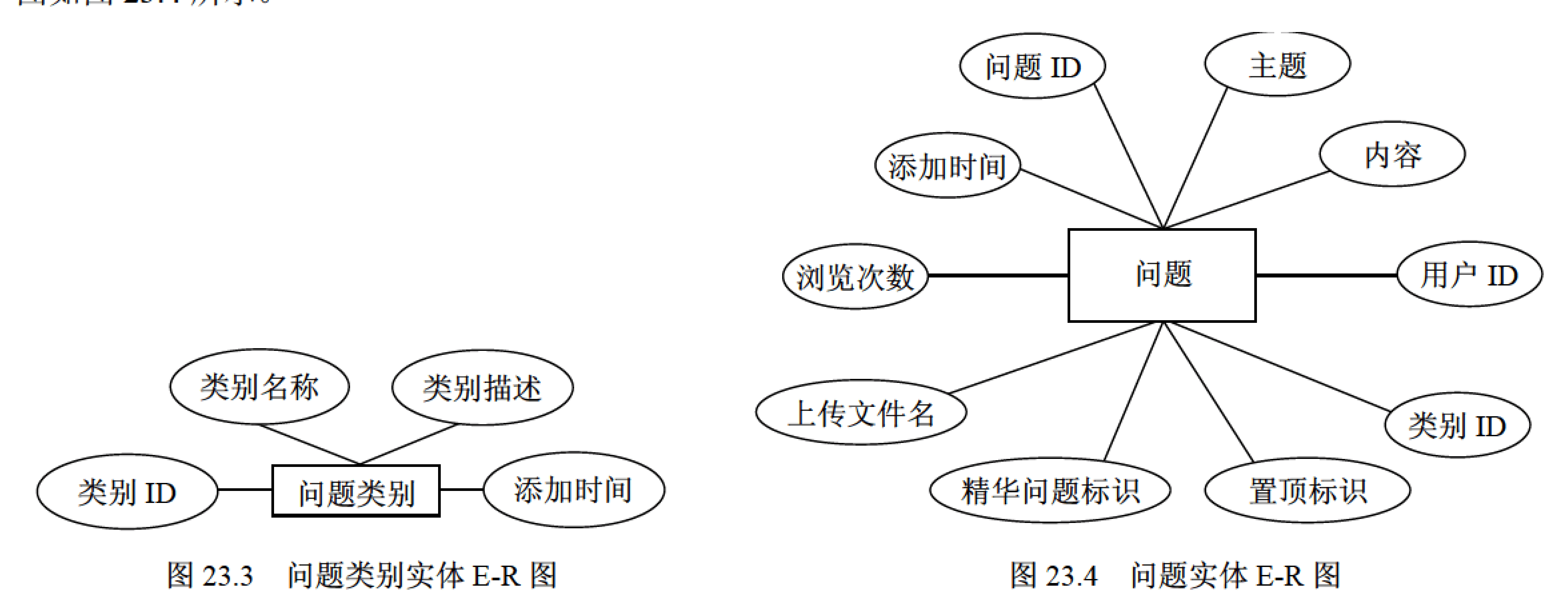


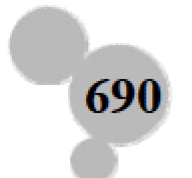
图 23.3　问题类别实体 E-R 图　　　图 23.4　问题实体 E-R 图

3. 回复实体

回复实体包括回复 ID 号、主题、内容、最佳答案标识、添加时间、用户 ID、回复问题的 ID 号、问题类别 ID 等属性，其实体 E-R 图如图 23.5 所示。

4. 关键字实体

关键字实体是对用户输入关键字进行记录的抽象，该实体包括关键字名称和关键字被检索次数等属性，其实体 E-R 图如图 23.6 所示。



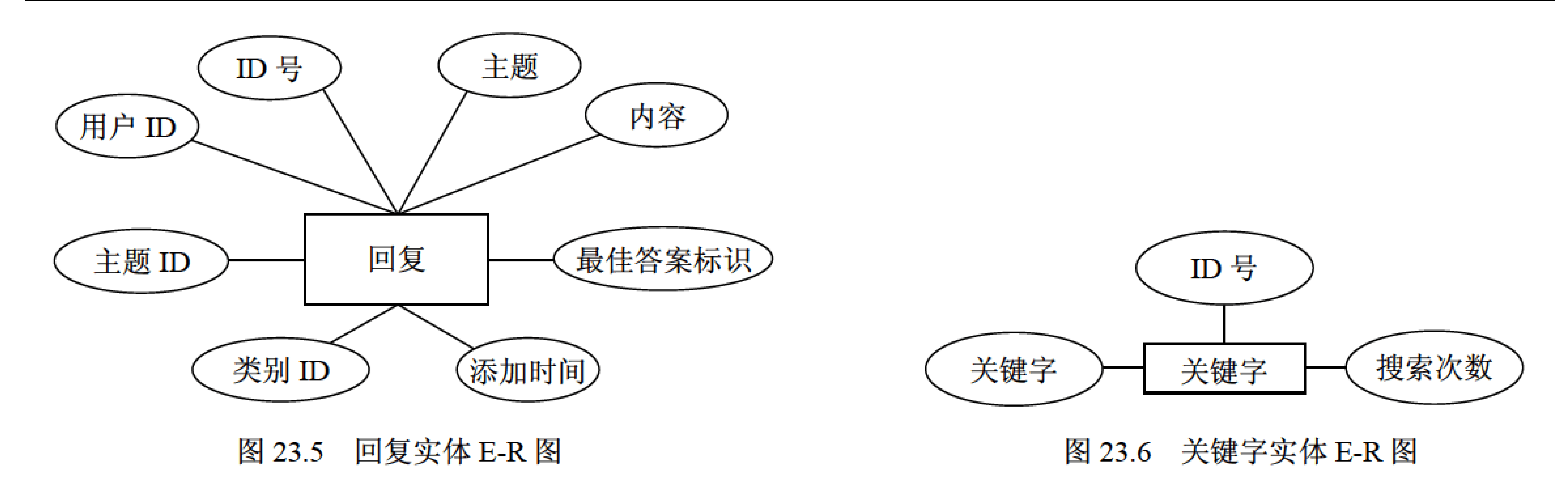


图 23.5　回复实体 E-R 图　　　　图 23.6　关键字实体 E-R 图

5. 用户实体

用户实体是对网站注册用户的抽象，其属性包括用户 ID、用户昵称、登录密码、E-mail、联系电话、所在地、注册时间、头像名称、发帖次数、回帖次数、积分、用户类别标识等，其实体 E-R 图如图 23.7 所示。

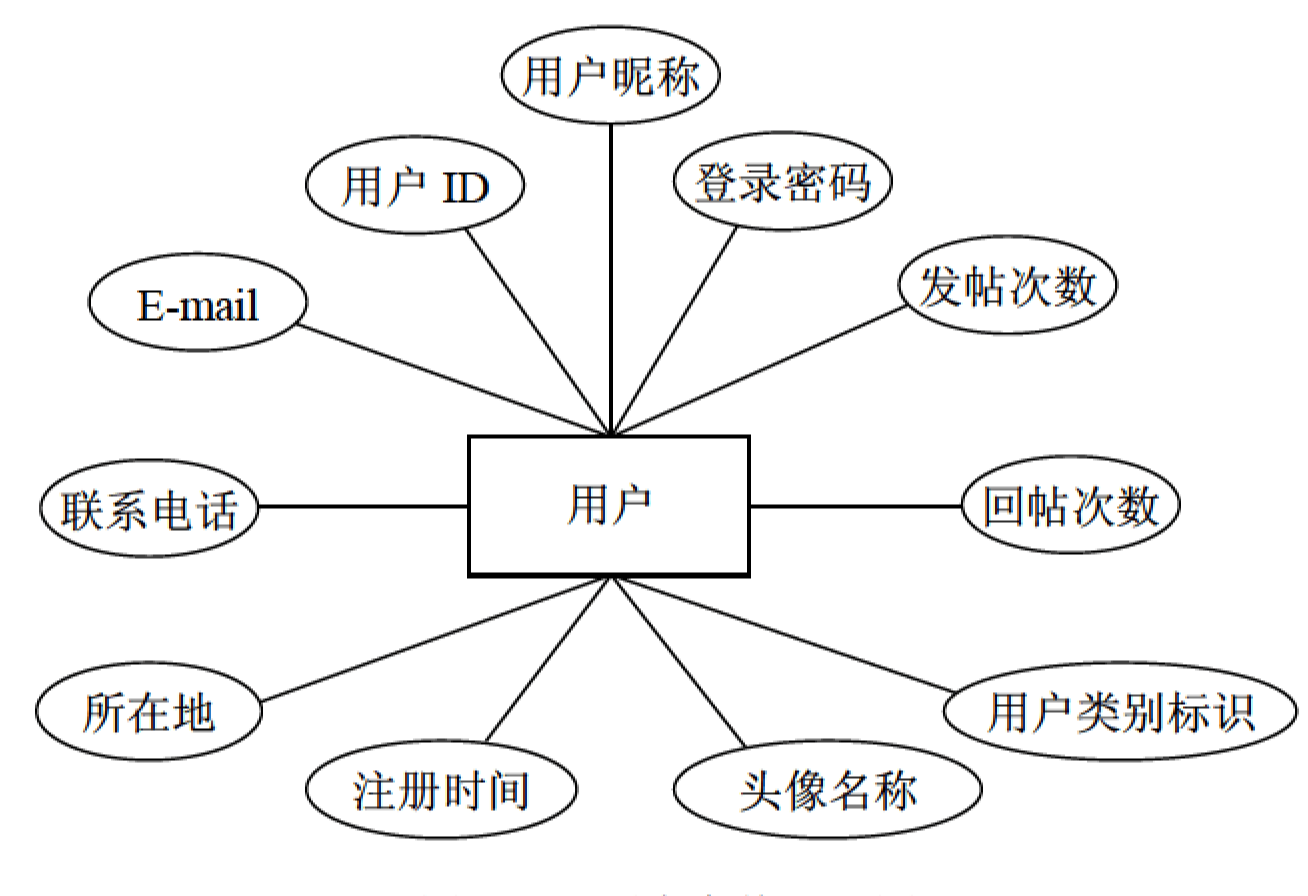


图 23.7　用户实体 E-R 图

23.5.3　创建数据库及数据表

根据需求分析及数据库概念设计，可以制定出“明日搜索引擎”网站数据库（这里命名为 db_database23）应该包括问题类别表（tb_bbstype）、问题表（tb_title）、回复表（tb_reply）、关键字表（tb_keywords）和用户表（tb_user）5 个表，其中各表的主要参数如图 23.8 所示。

下面将具体介绍“明日搜索引擎”数据库中各个数据表字段的具体描述。

1. 问题类别表（tb_bbstype）

问题类别表（这里命名为 tb_bbstype），用于存储问题类别，其结构如图 23.9 所示。

2. 问题表（tb_title）

问题表（这里命名为 tb_title），用于存储问题内容，其结构如图 23.10 所示。

3. 回复表（tb_reply）

回复表（这里命名为 tb_reply），用于存储用户对问题的回复，其结构如图 23.11 所示。

表	类型	整理	大小
tb_bbstype	MyISAM	utf8_unicode_ci	2.6 KB
tb_keywords	MyISAM	utf8_unicode_ci	4.5 KB
tb_reply	MyISAM	utf8_unicode_ci	4.7 KB
tb_title	MyISAM	utf8_unicode_ci	6.6 KB
tb_user	MyISAM	utf8_unicode_ci	10.8 KB

图 23.8　明日搜索引擎网站数据表主要参数

字段	类型	整理	属性	Null	默认	额外
id	int(11)			否		auto_increment
typename	varchar(50)	utf8_unicode_ci		否		
addtime	datetime			否		
description	varchar(255)	utf8_unicode_ci		否		

图 23.9　问题类别表结构

字段	类型	整理	属性	Null	默认	额外
id	int(11)			否		auto_increment
title	varchar(200)	utf8_unicode_ci		是	NULL	
content	text	utf8_unicode_ci		是	NULL	
addtime	datetime			是	NULL	
istop	tinyint(1)			是	NULL	
user_id	int(11)			是	NULL	
bbstype_id	int(11)			是	NULL	
browse	int(11)			是	NULL	
filename	varchar(50)	utf8_unicode_ci		是	NULL	
isjh	tinyint(1)			是	NULL	
unhtmlcontent	text	utf8_unicode_ci		否		

图 23.10　问题表结构

字段	类型	整理	属性	Null	默认	额外
id	int(11)			否		auto_increment
title	varchar(200)	utf8_unicode_ci		是	NULL	
content	text	utf8_unicode_ci		是	NULL	
addtime	datetime			是	NULL	
user_id	int(11)			是	NULL	
title_id	int(11)			是	NULL	
topindex	int(11)			是	NULL	
bbstype_id	int(11)			是	NULL	

图 23.11　回复表结构

4．关键字表（tb_keywords）

关键字表（这里命名为 tb_keywords），用于存储用户曾搜索的关键字及每个关键字被搜索的次数，其结构图如图 23.12 所示。

5．用户表（tb_user）

用户表（这里命名为 tb_user），用于存储用户信息，其结构图如图 23.13 所示。

字段	类型	整理	属性	Null	默认	额外
id	int(11)			否		auto_increment
keyword	varchar(50)	utf8_unicode_ci		否		
searchtime	int(11)			否		

图 23.12　关键字表结构

字段	类型	整理	属性	Null	默认	额外
id	int(11)			否		auto_increment
password	varchar(50)	utf8_unicode_ci		是	NULL	
netname	varchar(50)	utf8_unicode_ci		是	NULL	
email	varchar(100)	utf8_unicode_ci		是	NULL	
tel	varchar(20)	utf8_unicode_ci		是	NULL	
pc	varchar(20)	utf8_unicode_ci		是	NULL	
regtime	datetime			是	NULL	
face	varchar(50)	utf8_unicode_ci		是	NULL	
pubtimes	int(11)			是	NULL	
replytimes	int(11)			是	NULL	
usertype	int(11)			是	NULL	

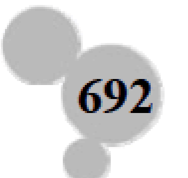
图 23.13　用户表结构

23.6　技　术　攻　关

“明日搜索引擎”网站在开发过程中，多处使用了较新的技术。为了便于对该项目各个模块的理解，本节首先对“明日搜索引擎”网站所使用的关键技术进行讲解。

23.6.1　类似 Google 搜索引擎的搜索条

为了方便网站浏览者检索网站内容，很多知名网站的查询关键字文本框都提供了关键字提示下拉列表



框，这样用户就可以根据以往自己所输入的关键字更加准确地定制自己的关键字，从而在很大程度上提高了要查询内容被高效、准确查询出来的几率。同样，在制作“明日搜索引擎”网站的搜索模块中也引用有关键字提示的搜索条，如图 23.14 所示。

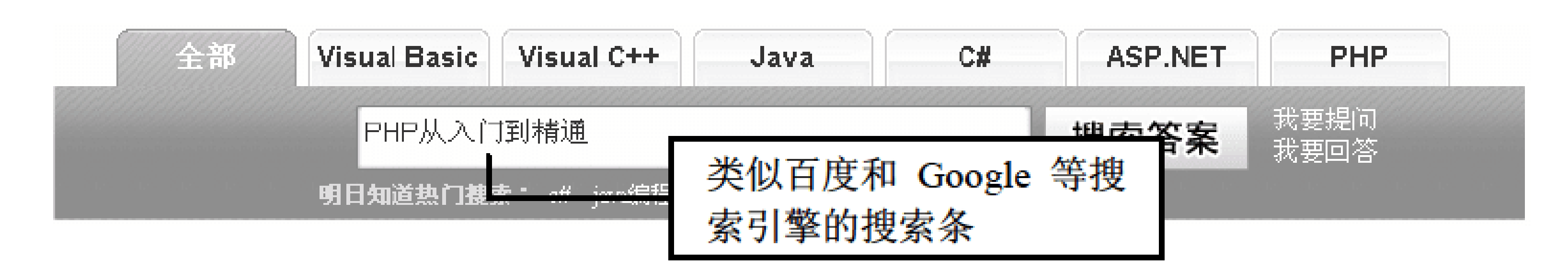


图 23.14 类似 Google 搜索引擎的搜索条

那么，类似 Google 的搜索条是如何实现的呢？是不是特别复杂呢？继续向下学习将会寻求到答案。在具体讲解实现过程前，首先讲解制作该搜索条的思路，当用户在关键字文本框中输入内容时，通过文本框的 onKeyUp 事件调用来显示关键字下拉列表的方法，该方法通过 Ajax 技术（这里使用 Ajax 的框架 jQuery）应用 GET 方法向服务器发送请求，请求过程中将用户输入的关键字发送给服务器，服务器经过处理返回所有以该关键字开头的用户曾搜索的关键字及搜索次数和与该关键字匹配的结果个数。下面具体讲解类似 Google 搜索条的实现方法。

（1）建立用于录入关键字的文本框，这里使用 Zend Framework 框架的 formText()视图助手建立。Zend Framework 框架的视图助手，大部分具有用来生成组件和有自动转义变量的功能。另外，有些助手用来创建基于路由的 URL 和 HTML 列表以及声明变量。这里所使用的 formText()助手用来生成 HTML 的文本框组件。

formText()视图助手的语法格式如下：

```
formText(string name, string value, array attribs)
```

formText()视图助手的参数说明如表 23.1 所示。

表 23.1 formText()视图助手的参数说明

参 数	说 明
name	必要参数，指定所生成文本框组件的名称
value	可选参数，文本框的内容，默认为 null
attribs	可选参数，以数组形式指定文本框的多个属性，默认为 null

“明日搜索引擎”网站使用 formText()视图助手生成搜索文本框的代码如下：

```
echo $this->formText('keywords', '', array('style'=>'width:350px; height:26px; line-height:26px; font-size:15px; color:#333333; border:0px;', 'onKeyUp'=>'showKeywordsList()', 'onKeyDown'=>'selectList()', 'autocomplete'=>'Off'))                    //通过 formText()视图助手输出查询关键字录入文本框
```

上述代码中，将关键字文本框命名为 keywords，默认值为空串，在属性数组中，通过键名为 style 的数组元素指定文本框样式，通过键名为 onKeyUp 的数组元素指定当按键抬起时所触发的事件，通过键名为 onKeyDown 的数组元素指定当按键按下时所触发的事件，最后为键名为 autocomplete 的数组元素指定 Off 值来去除文本框自身的下拉列表提示。

（2）当用户在文本框中录入关键字时，将调用自定义的 showKeywordsList()方法显示关键字下拉列表。代码如下：

代码位置：光盘\MR\23\application\modules\default\views\scripts\index\index.phtml

```
var index = 0;                                  //关键字列表项索引
var count = 0;                                  //关键字列表总数
function showKeywordsList(){                    //显示关键字下拉列表
    var keycode = event.keyCode;                //用户按键的 ASCII 码
```

```
    if(keycode != 40 && keycode != 38){                    //如果不是向上和向下方向键
        index = 0;                                         //关键字索引归 0
        if($("#keywords").val() == "" || $.trim($("#keywords").val()) == ""){
            $("#searchLayer").css("display", "none");      //如果用户输入空字符或空格则隐藏下拉列表
        }else{
❶               setTimeout("reSearch()", 150);             //否则每隔 150 毫秒调用一次 reSearch()方法
        }
    }
}
```

代码导读：

❶ 这里之所以要间隔 150 毫秒后调用 reSearch()方法，是为了降低连续对服务器请求而带来的负载。

上述代码中，首先使用 JavaScript 的 event 对象的 keyCode 属性获取用户输入字符的 ASCII 码，然后判断该 ASCII 码是否为 40（代表向上方向键）或 38（代表向下方向键），如果不是这两个方向键，并且用户录入的字符不为空串或空格，则间隔 150 毫秒调用 reSearch()方法来显示与用户录入关键字所匹配的关键字搜索记录。其中 reSearch()方法的代码如下：

代码位置：光盘\MR\23\application\modules\default\views\scripts\index\index.phtml

```
function reSearch(){
if($("#keywords").val() != ""){                            //如果用户录入的关键字不为空
//通过 jQuery 向服务器发送 GET 请求
❶       $.get("<?php echo $this->baseUrl('/index/keywords-list/keyword/" +encodeURI($("#keywords"). val())+"')?>",
null, function(data){
            if($.trim(data) == ""){                        //如果返回数据为空，则隐藏下拉列表层
                $("#searchLayer").css("display", "none");
            }else{
                $("#searchLayer").html(data);              //如果返回数据不为空则显示下拉列表
                $("#searchLayer").css("display", "block");
❷           count = parseInt($("#totalList").val());       //获取列表中关键字个数
            }
        });
    }
}
```

代码导读：

❶ encodeURI()方法用来对关键字进行 URL 编码。

❷ parseInt()方法用来将字串转换为整型。

上述代码中，通过 jQuery 的$.get()方法向服务器发送请求，如果返回结果不为空串，则将内容显示在下拉列表中。

说明：通过 Ajax 技术向服务器发送请求，传递的内容应该为 UTF-8 编码，如果不是，应该使用函数 iconv()对传递的内容进行转码。

（3）当用户移动键盘的上下方向键时，列表中所选中的项以特殊颜色显示，并在关键字文本框中显示当前选中项的内容。代码如下：

代码位置：光盘\MR\23\application\modules\default\views\scripts\index\index.phtml

```
function selectList(){
    var keycode = event.keyCode;                           //获取用户录入字符的 ASCII 码
    if(keycode == 40){                                     //如果用户按向下方向键
```

```
        $("#listItem_"+index).css("background-color", "#158CD0");        //更改当前所选项的背景色
        $("#listItem_"+index).css("color", "#FFFFFF");                   //更改当前所选项的前景色
        $("#keywords").val($.trim($("#listItem_li_"+index).html()));     //将当前所选项显示在文本框中
        if(index > 0){
            $("#listItem_"+parseInt(index-1)).css("background-color", "#FFFFFF"); //更改前一项的背景色
            $("#listItem_"+parseInt(index-1)).css("color", "#333333");    //更改当前项前一项的前景色
            }
            if(index < count-1){                                         //如果当前索引小于总关键字数减 1
             index++;                                                    //索引数增 1
            }
       }else if(keycode == 38){                                          //如果用户按向上方向键
        if(index > 0){                                                   //如果索引大于 0
              $("#listItem_"+parseInt(index-1)).css("background-color", "#158CD0"); //更改当前项前一项的背景色
              $("#listItem_"+parseInt(index-1)).css("color", "#FFFFFF");        //更改当前项前一项的前景色
              $("#keywords").val($.trim($("#listItem_li_"+parseInt(index-1)).html()));      //将选项显示在文本框中
        }
             $("#listItem_"+index).css("background-color", "#FFFFFF"); //更改当前项的背景色
             $("#listItem_"+index).css("color", "#333333");             //更改当前项的前景色
            if(index > 1){                                               //如果索引大于 1
             index--;                                                    //索引减 1
            }
       }
}
```

上述代码首先获取用户输入字符的 ASCII 码，然后判断该值是否为向上或向下方向键所对应的 ASCII 码，如果是则通过 jQuery 的 css()方法改变所选项的前景色和背景色，从而达到关键字列表中内容上下切换的效果。

（4）在上面的讲解中，已经提及客户端通过 GET 方法向服务器发出请求，那么发出请求后做了哪些处理呢？从下述代码中将会找到答案。

代码位置：光盘\MR\23\application\modules\default\controllers\IndexController.php

```
public function keywordsListAction ()
{
❶   header('content-type:text/html; charset=utf-8');                  //设置页面编码为 utf-8
    $keyword = urldecode($this->_request->getParam('keyword'));  //获取用户录入的关键字并进行 URL 解码
❷   $keywordses = $this->_keywordsModel->findByLike($keyword, 10, true);
                                         //调用关键字模型的 findByLike()方法查询所有以该关键字开头的内容
    $total = 0;                                                      //保存总匹配关键字数
    if ($keywordses != null) {
          $total = count($keywordses);                               //统计总匹配关键字数
    }
    $str = '<input type="hidden" name="totalList" id="totalList" value="' . $total . '" />';
    foreach ($keywordses as $k => $key) {                            //通过循环形成下拉列表
          $str  .= '<div id="listItem_' . $k . '" style="width:98%; clear:both; background-color:#FFFFFF; cursor:
pointer;" onmouseover="this.style.backgroundColor=\'#158CD0\'; this.style.color=\'#FFFFFF\'" onmouseout ="this.
style.backgroundColor=\'#FFFFFF\'; this.style.color=\'#333333\'" onclick="setText(\'' . $key['keyword'] . '\')">';
          $str .= '<ul>';
          $str .= ' <li id="listItem_li_' . $k . '" style="width:60%; height:18px; line-height:18px; text-align:left; float:left;">' .
$key['keyword'] . '</li>';
          $str .= ' <li style="width:38%; height:18px; line-height:18px; text-align:right; float:right;">' . $key['searchtime'] .
'</li>';
```

```
            $str .= '</ul>';
            $str .= '</div>';
        }
        echo $str;
        $this->_helper->layout->disableLayout();                    //取消页面布局
        $this->_helper->viewRenderer->setNoRender();                //取消视图
}
```

代码导读：

❶ header()函数用于向客户端浏览器发送 HTTP 头信息，这里用来指定页面所使用的编码为 UTF-8。

❷ 这里调用关键字模型的 findByLike()方法，获取关键字被搜索最多的 10 条记录。

上述代码为 default 模块中，IndexController 控制器的 keywordsListAction 动作，主要用来获取下拉列表的数据并发送给客户端。

由于上述代码只需要显示下拉列表内容，不需要显示网站头和尾信息，即不需要使用布局，所以使用如下代码取消该动作的页面布局：

```
$this->_helper->layout->disableLayout();
```

注意：使用 header()函数发送 HTTP 头应该在所有页面输出之前。

Zend Framework 中，使用当前控制器对象的_help 属性的 layout 属性的 disableLayout()方法取消布局。

同样，由于不需要视图，直接通过 echo 语句就可以输出列表中的内容，所以可以将该 Action 所使用的视图通过如下代码取消。

```
$this->_helper->viewRenderer->setNoRender();
```

注意：由于“明日搜索引擎”网站使用 Zend Framework 框架开发，学习本堂课内容需要有一定的 MVC 开发思想和 Zend Framework 框架基础。

23.6.2 使用空格分隔多关键字

用户在使用百度、Google 等互联网搜索工具时，如果对多个关键字同时进行搜索，可以将多个关键字用空格分隔，从而提高查找效率。

“明日搜索引擎”网站中为了方便用户查找，采用了分词技术可以对多个关键字进行同时查找。例如，用户输入“编程”和“软件”两个关键字，并且关键字用空格分隔，提交搜索后将查找出所有包含这两个关键字的内容，如图 23.15 所示。

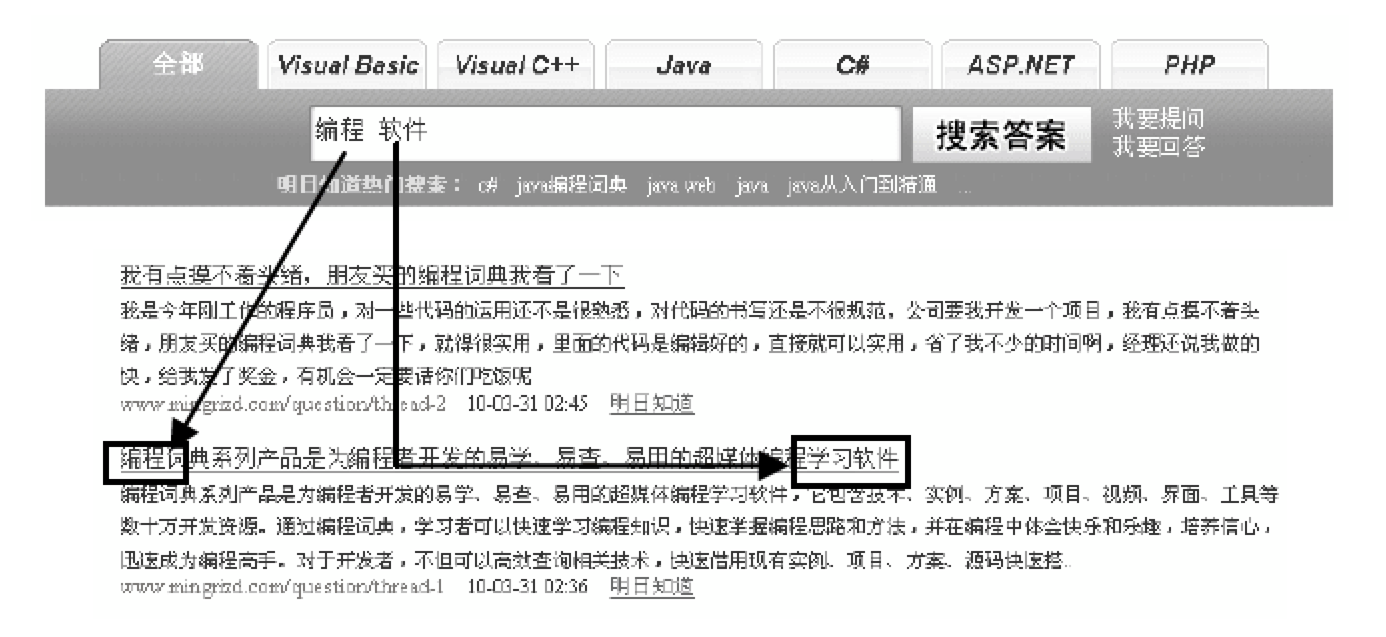


图 23.15 多关键字同时查询

下面讲解这种实用的分词技术的实现方法。



（1）当用户输入完关键字提交查询内容后，首先在 IndexController 控制器的 searchAction 动作方法中使用 getParam()方法获取用户提交关键字的值，代码如下：

```
$keywordsStr = urldecode($this->_request->getParam('keywords'));
```

你问我答：Zend Framework 接收 POST 或 GET 传递参数的方法。

Zend Framework 框架中接收页面间传递参数的方法有 3 种，分别为当前控制器对象_request 属性的 getQuery()方法、getPost()方法和 getParam()方法。这 3 种方法的区别是：getQuery()方法用来获取 GET 方法提交的数据、getPost()方法用来获取 POST 方法提交的数据，而通过 getParam()方法既可以获取 GET 方法提交的数据，还可以获取 POST 方法提交的数据。

（2）在主题表（tb_title）模型 Model_DbTable_Title 中建立 findByLike()方法，用来对用户输入的关键字进行模糊查找。代码如下：

代码位置：光盘\MR\23\application\models\DbTable\Title.php

```
public function findByLike ($keywordsStr, $bbstype_id = 0, $orderFlag = null, $page = 1, $pageSize = 10, $pageRange = 5)
{
❶   $arrayKeyWords = explode(' ', $keywordsStr);                //将多个关键字用空格分割保存到数组中
    if ($orderFlag == null || $orderFlag == 0) {                //指定排序方式
        $order = 'addtime desc';                                //时间降序
    } else {
        $order = 'browse desc';                                 //时间升序
    }
    $select = $this->getAdapter()->select();                    //生成 Zend_Select 对象
    $select->from($this->_name);                                //为 Zend_Select 对象指定表名
    foreach ($arrayKeyWords as $keywords) {                     //遍历所有被空格分隔的关键字
        if (trim($keywords) != '') {                            //如果关键字不为空串
❷$select->orWhere($this->getAdapter()->quoteInto('title like ?', '%' . $keywords . '%'));
//对主题进行匹配查询
            if ($bbstype_id != 0) {
                $select->where('bbstype_id = ' . $bbstype_id);  //指定查询类别
            }
            $select->orWhere($this->getAdapter()->quoteInto('unhtmlcontent like ?', '%' . $keywords . '%'));
                                                                //对内容进行匹配查询
            if ($bbstype_id != 0) {
                $select->where('bbstype_id = ' . $bbstype_id);  //指定查询类别
            }
        }
    }
    $select->order($order);                                     //指定查找结果排序方式
    $adapter = new Zend_Paginator_Adapter_DbSelect($select);    //指定分页适配器
    $paginator = new Zend_Paginator($adapter);                  //生成 Zend_Paginator 分页对象
$paginator->setItemCountPerPage($pageSize)->setCurrentPageNumber($page)->setPageRange($pageRange);
                                                                //指定分页参数
    return $paginator;                                          //将 Zend_Paginator 对象作为结果返回
}
```

代码导读：

❶ explode()函数用来将字符串用指定子串分隔，并保存到数组中。

❷ $this->getAdapter()->quoteInto()方法用来防止非法字符注入。

（3）在 searchAction 动作方法中通过 tb_title 表模型实例的对象调用 findByLike()方法来获取 Zend_Paginator 对象对查询结果进行分页显示。代码如下：

```
$paginator = $this->_titleModel->findByLike($keywordsStr, $typeid, null, $page, $pageSize);
```

（4）在 searchAction 动作的视图中，通过 foreach 循环语句输出查询结果。代码如下：

```
<?php foreach ($this->paginator as $search):?>
<!--省略 HTML 代码，用来显示查询结果-->
<?php endforeach;?>
```

（5）为了将查询结果进行分页显示，还需要建立分页导航。Zend_Paginator 的分页控制器对象的视图提供了多个便于开发人员建立分页的属性，如表 23.2 所示。

表 23.2　Zend_Paginator 分页控制器对象的视图常用属性

属　　性	说　　明
first	首页页码号
firstItemNumber	当前页第一条记录绝对编号
firstPageInRange	分页导航中首页页码号
current	当前页的页号
currentItemCount	当前页记录总数
itemCountPerPage	每页显示的记录数
last	尾页页码号
lastItemNumber	最后一条记录号
lastPageInRange	分页导航中最后一页页码号
next	下一页页号
pageCount	总页数
pagesInRange	根据所使用的导航条样式，返回页码数组
previous	上一页页码号
totalItemCount	记录总数

了解分页控制器视图的主要属性后，下面开始讲解建立分页导航的步骤。首先需要在 search.phtml 视图文件中所要显示视图导航的位置通过视图对象的 paginationControl()方法指定分页导航的属性，代码如下：

```
<?php echo $this->paginationControl($this->paginator, 'Sliding' ,'searchPaginationControl.phtml', array('keywords' =>
urlencode($this->keywordsStr)))?>
```

视图对象的 paginationControl()方法的语法格式如下：

```
paginationControl(Zend_Paginator paginator, string style, string paginationController, array params)
```

视图对象的 paginationControl()方法的参数说明如表 23.3 所示。

表 23.3　视图对象的 paginationControl()方法的参数说明

参　　数	说　　明
paginator	必要参数，指定 Zend_Paginator 对象
style	可选参数，指定导航条样式，具体样式说明请参见表 23.4
paginationController	可选参数，指定分页控制器视图文件名
params	可选参数，以数组形式指定要传递给分页控制器视图变量名称和值

表 23.4　导航条样式说明

参　数	说　明
All	显示所有页码链接
Elastic	类似 Google 所使用的导航条
Sliding	类似 Yahoo 所使用的导航条

最后还需要建立导航视图，“明日搜索引擎”网站中导航视图的代码如下：

代码位置：光盘\MR\23\application\modules\default\views\scripts\searchPaginationControl.phtml

```
<ul>
    <li class="li1">每页 <?php echo $this->itemCountPerPage?> 条/共查找到 <?php echo $this->totalItemCount?> 条</li>
    <li class="li_cell"></li>
    <?php if($this->totalItemCount>0):?>
        <?php foreach ($this->pagesInRange as $page):?>
    <li onclick="javascript:window.location.href='<?php echo $this->baseUrl('/index/search/keywords/'.$this->keywords.'/page/'.$page)?>';" class="<?php if($this->current==$page):?>li2_current<?php else:?>li2<?php endif;?>"><a href="<?php echo $this->baseUrl('/index/search/keywords/'.$this->keywords.'/page/'.$page)?>" class="<?php if($this->current==$page):?>a2<?php else:?>a1<?php endif;?>"><?php echo $page?></a></li>
    <li class="li_cell"></li>
        <?php endforeach;?>
        <?php if($this->last > 5):?>
    <li onclick="javascript:window.location.href='<?php echo $this->baseUrl('/index/search/keywords/'.$this->keywords.'/page/'.$this->last)?>';" class="li3"><a href="<?php echo $this->baseUrl('/index/search/keywords/'.$this-> keywords.'/page/'.$this->last)?>" class="a1">...<?php echo $this->last?></a></li>
        <?php endif;?>
    <?php endif;?>
</ul>
```

用户使用“明日搜索引擎”网站查询内容，将在查询结果页面显示如图 23.16 所示的分页导航条。

每页 20 条/共查找到 110 条　1　2　3　4　5

图 23.16　多关键字同时查询

23.6.3　关键字描红

使用互联网搜索引擎时，当用户在搜索条中输入完关键字并提交查询信息后，会在查询结果显示页面查看到，所有输入的查询关键字都以红色显示，其实质就是关键字的描红输出。通过关键字的描红技术，可以使查询结果更醒目，便于用户进一步筛选。“明日搜索引擎”网站的查询结果描红输入的效果如图 23.17 所示。

图 23.17　查询结果的关键字描红输出效果

下面将介绍 PHP 中搜索模块关键字描红输出的制作方法。在详细介绍描红的制作步骤前，首先介绍一下描红的实现思路：当用户提交查询关键字到搜索页面后，通过当前控制器对象的_helper 属性的 getParam()方法获取用户提交关键字，然后使用 PHP 提供的 explode()函数将提交关键字以空格作为分界符保存到数组中，并将该数组保存到视图变量中，最后定义一个视图助手，用于将该数组中所有关键字描红输出。具体实现过程如下：

（1）接收用户提交的关键字，并将关键字保存到数组中，然后将该数组赋给视图变量，以便在视图层能够使用该数组，代码如下：

```
$keywordsStr = urldecode($this->_request->getParam('keywords'));    //接收用户提交的查询关键字
$keywordsStrArray = explode(' ', $keywordsStr);                     //将关键字字符串用空格分隔保存到数组中
$this->view->assign('keywordsStrArray', $keywordsStrArray);        //将数组赋给视图变量 keywordsStrArray
```

（2）在 views 文件夹下的 helpers 子文件夹下建立 SetRed.php 文件，在该文件中建立用于将关键字描红输出的自定义助手 Zend_View_Helper_SetRed。代码如下：

代码位置：光盘\MR\23\application\modules\default\views\helpers\SetRed.php

```
class Zend_View_Helper_SetRed
{
    public function setRed ($arrayKeywords, $str)
    {
        $beginFont = '︼';                                          //关键字开始标识
        $endFont = '︻';                                            //关键字结束标识
        foreach ($arrayKeywords as $keyword) {                      //遍历所有关键字
            $array = array();
❶           preg_match_all('/' . $keyword . '/i', $str, $array);   //查询与关键字匹配的内容，并保存到数组中
            $arrayKey = array_unique($array[0]);                    //去掉重复数组元素
            foreach ($arrayKey as $key) {                           //遍历所有匹配关键字
//将查询结果中匹配关键字前后分别加上关键字开始标识和结束标识
                $str = str_replace($key, $beginFont . $key . $endFont, $str);
            }
        }
        $str = str_replace($beginFont, '<font color="#FF0000">', $str); //将开始标识换成<font>开始部分
        $str = str_replace($endFont, '</font>', $str);              //将结束标识替换成<font>标签结束部分
        return $str;
    }
}
```

代码导读：

❶ preg_match_all()函数的语法格式为 preg_match_all(string pattern, string subject, array matches, [.int flags])，该函数用于在 subject 中搜索与 pattern 给出的正则表达式匹配的内容并将结果以 flags 指定的顺序放到 matches 中。

上述所定义的视图助手包含关键字数组和要描红的字符串两个参数。该视图助手首先通过 foreach()语句遍历关键字数组，然后使用 preg_match_all()函数提取要描红字符串中所有与关键字相匹配的字符串，并将匹配的字符串保存在一个数组中，去掉重复元素后，将该数组所有元素的开始和结束加上一个标识，最后将所有开始标识替换为，结束标识替换成。最终达到在所要描红字符串的前后加上<font>标签，这样可以达到在输出该字符串时，字符串中的匹配关键字描红输出的效果。

📖 **说明：**Zend Framework 框架中，自定义视图助手一般存放在 views 目录下的 helpers 子目录中，视图助手文件应以大写字母开头，视图文件中的类名要以 Zend_View_Helper_作为前缀，然后紧接与视



图文件名相同的字符作为完整类名，视图文件中包含一个与视图文件名相同的方法，但该方法首字母要小写。

（3）在 search.phtml 文件中通过上面介绍的试图助手将查询结果中的关键子进行描红输出。代码如下：

```
echo $this->setRed($this->keywordsStrArray, htmlspecialchars($this->substr($search['unhtmlcontent'], 500)))
```

23.6.4　制作在线编辑器

以往使用 HTML 提供的文本域对文本进行内容编辑，一般是通过 JavaScript 在要编辑文本的前后增加一些标记，然后在输出这些内容时，将这些标记替换成相应的 HTML 标记（如<font>、<style>等），这样在显示内容时，就可以出现相应的特效。但是，这种编辑器并不能达到所见即所得的效果，目前已经有很多组织为开发人员提供了在线文本编辑工具，而且功能也非常强大，其中 FCKEditor 就是较著名的在线编辑器之一，开发人员只需简单的配置，就可以在项目中应用该编辑工具，不过这些编辑器由于功能较强大，可能存在访问页面时加载速度慢、针对性不强等缺陷。“明日搜索引擎”网站在发表问题和回复问题所使用的在线编辑器是通过 JavaScript 编写的，具备文字排版、更改文字样式、插入文字链接和插入图片等功能，完全适合论坛、博客和后台管理中的内容编辑工作。如图 23.18 所示为“明日搜索引擎”网站所使用的在线编辑器。

图 23.18　在线编辑器

在讲解在线编辑器具体实现步骤前，首先讲解制作一个在线编辑器的基本思路。“明日搜索引擎”网站所使用编辑器的编辑区域实质为一个<iframe>内嵌标签，默认情况下<iframe>标签是不可编辑的，需要设置<iframe>的 document 属性的 designMode 属性的值为 on，来开启该标签的可编辑模式，然后使用<iframe>的 document 属性的 executeCommand()方法执行相应的命令标识即可，其中常用的命令标识如表 23.5 所示。

表 23.5　常用命令标识

参　　数	说　　明
BackColor	设置或获取当前选中区的背景颜色
Bold	切换当前选中区的粗体显示与否
CreateLink	在当前选中区上插入超链接，或显示一个对话框允许用户指定要为当前选中区插入的超链接的 URL
Cut	将当前选中区复制到剪贴板并删除
Paste	用剪贴板内容覆盖当前选中区
Copy	将当前选中区复制到剪贴板
Delete	删除当前选中区
FontName	设置或获取当前选中区的字体
FontSize	设置或获取当前选中区的字体大小
ForeColor	设置或获取当前选中区的前景（文本）颜色
Italic	切换当前选中区斜体显示与否
JustifyCenter	将当前选中区所在格式化块置中
JustifyLeft	将当前选中区所在格式化块左对齐
JustifyRight	将当前选中区所在格式化块右对齐
InsertImage	用图像覆盖当前选中区
Undo	取消
Redo	重做

续表

参　数	说　明
underline	切换当前选中区的下划线显示与否
strikethrough	删除线

了解在线编辑器的基本思路后，下面将具体讲解在线编辑器的制作步骤。

（1）建立在线编辑器 UI。在线编辑器是通过 JavaScript 实现的，所以将 HTML 代码作为字符串，然后使用 JavaScript 的 document 对象的 write()方法输出到浏览器即可，由于代码所占篇幅较长，这里只给出关键代码，如下所示。

代码位置：光盘\MR\23\public_html\js\LzhEditor\LzhEditor.js

```
LzhEditor.prototype.Create = function() {
    this.editorStr += "<div id=\"faceLayer\" style=\"position: absolute; width:445px; z-index: 1; border:1px solid #77B7DD; background-color:#F2F9F9; clear:both;   display:none;\" onmouseleave=\"this.style.display='none'\">";
    …//省略代码请详见本书附带光盘，该段代码主要用于构建编辑器的控制面板
    this.editorStr += "<iframe id=\"editor\" name=\"editor\" width=\"100%\" height=\""+ this.height + "\" scrolling=\"auto\" frameborder=\"0\"></iframe>";                //通过 iframe 建立编辑区域
    this.editorStr += "<input type=\"hidden\" id=\"" + this.fieldName+ "\" name=\"" + this.fieldName + "\" value=\"\" />";                //通过隐藏域保存用户在编辑器录入的内容
    this.editorStr += "</div>";
    document.write(this.editorStr);
    window.frames["editor"].document.open();                //打开 iframe
    window.frames["editor"].document.write("<BODY style=\"PADDING-RIGHT: 5px; PADDING-LEFT: 5px; FONT-SIZE: 12px; PADDING-BOTTOM: 5px; MARGIN: 0px; PADDING-TOP: 5px\">"+ this.value + "</BODY>");
//将<body>标签写入 iframe
    window.frames["editor"].document.close();                //关闭 iframe
❶   window.frames["editor"].document.designMode = "on";     //开启 iframe 的编辑模式
    window.frames["editor"].focus();                        //iframe 获得焦点
    window.frames["editor"].document.onKeyDown = function() {    //为 iframe 添加 onKeyDown 事件
        if (window.frames["editor"].event.keyCode == 13) {       //如果是回车
            window.frames['editor'].document.selection.createRange().pasteHTML('<br><!---->');
 //<br>后必须有内容才能换行，所以<br>后又加了 HTML 的注释，这样既可以实现换行，还不会显示多余内容
            window.frames['editor'].event.returnValue = false;
        }
    };
};
```

代码导读：

❶ 通过设置<iframe>标签 document 属性的 designMode 属性的值为 on，来开启<iframe>标签的编辑模式。

上述代码中，使用<iframe>标签作为在线编辑器的编辑区域，默认情况下<iframe>标签是不可编辑的，所以需要开启<iframe>标签的编辑模式。在开启<iframe>标签的编辑模式前，需要将<body>标签写入到<iframe>内部来设置编辑区域边框间距、文字大小等属性。向<iframe>中写入<body>标签，首先需要调用<iframe>标签的 document 属性的 open()方法打开<iframe>，然后再通过该属性的 write()方法将<body>标签写入到<iframe>内部，最后使用该属性的 close()方法关闭<iframe>标签。

（2）建立完在线编辑器的躯壳后，下一步就需要为编辑器注入灵魂。注入灵魂的过程实质就是让在线编辑器控制面板真正具有排版、设置字体样式等功能，即为控制面板中各个图标按钮的 onClick()事件设置相应的方法。为了提高代码重用率，这里定义一个名为 LzhEditorFormat()的方法，该方法中有两个参数，分

别为要执行的命令标识（即表 23.5 所列出的命令标识）及执行该标识所带的参数。该方法的代码如下：

代码位置：光盘\MR\23\public_html\js\LzhEditor\LzhEditor.js

```
function lzhEditorFormat(hc, pa) {
    window.frames["editor"].focus();                                        //编辑器获得焦点
    window.frames["editor"].document.selection.createRange();               //创建编辑区域
    if (pa == "") {
        window.frames["editor"].document.execCommand(hc, false);            //不需要指派参数
    } else {
        window.frames["editor"].document.execCommand(hc, false, pa);        //需要指派参数
    }
}
```

通过上述代码可知，通过<iframe>标签的 document 属性的 execCommand()方法执行命令标识，该方法的语法格式如下：

```
bSuccess = object.execCommand(sCommand [, bUserInterface] [, vValue])
```

execCommand()方法的参数说明如表 23.6 所示。

表 23.6　execCommand()方法的参数说明

参　　数	说　　明
sCommand	必要参数，所要执行的命令标识
bUserInterface	可选参数，boolean 型，默认为 FALSE，即不显示用户接口
vValue	可选参数，执行命令标识，所带参数。如设置字体大小时，字号通过该参数指定

（3）通过上述两步讲解，相信大家已经了解了制作一个在线编辑器的基本思路，最后还需要讲解如何在程序中调用该编辑器。

由于该编辑器被封装在一个独立的 js 文件中，所以首先应该通过<script>标签的 src 属性包含该 js 文件。然后在 form 表单要显示编辑器的位置使用如下代码引入编辑器。

```
var lzhEditor = new LzhEditor(name, width, height ,value, baseUrl, uploadFileUrl);    //实例编辑器
lzhEditor.Create();                                                                    //创建编辑器
```

通过上述代码就可以将编辑器显示在表单区域内，但当用户提交表单前，还需要通过如下代码将编辑器中的内容赋给隐藏域，表单提交时不能直接将编辑器中的内容提交给服务器，而是间接地通过隐藏域实现。代码如下：

```
lzhEditor.Submit();                                        //将编辑器中的内容赋给隐藏域
```

以上就是制作一个在线编辑器的全部过程，最后总结其开发思路：首先开启<iframe>标签的编辑模式，然后通过 execCommand()方法执行对应的命令标识，即可对<iframe>中的文本进行编辑。

23.7　MVC 框架结构搭建

“明日搜索引擎”网站采用 Zend Framework 框架技术开发，所以首先应该设计并搭建出稳定、扩展性强的框架结构。

说明：有关使用 Zend Framework 建立多模块 MVC 框架结构的过程和方法已经在第 18 堂课的 18.2 节中进行了详细的讲解，这里不再赘述。

下面直接给出本项目中搭建的 MVC 框架结构，如图 23.19 所示。从图中可以看出，该结构是一种多模

块结构，即将前台和后台独立成两个模块，从而更有利于多人协作开发。

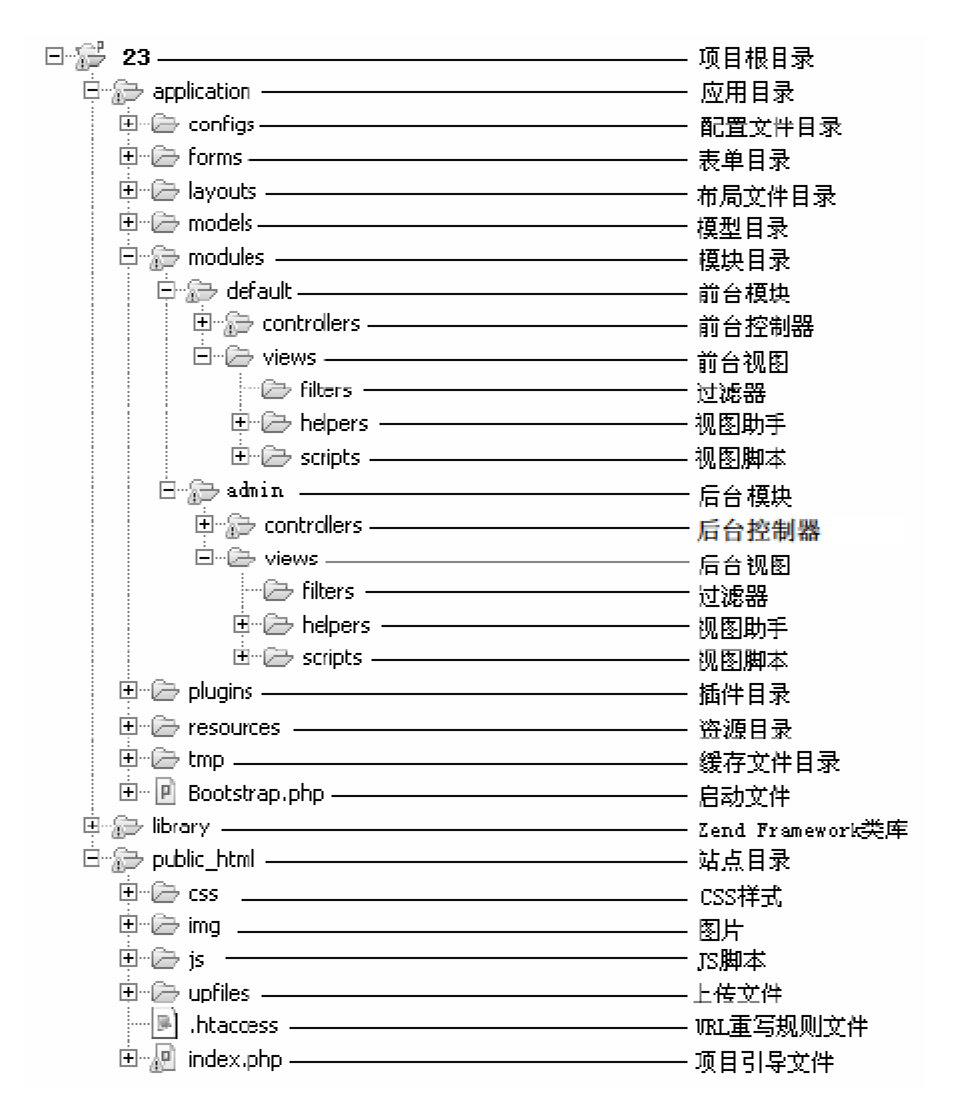


图 23.19　明日搜索引擎网站框架结构

23.8　首 页 设 计

网站首页的内容应该突出网站主题及特色，在网站的生命周期中起着非常重要的作用，本节将以“明日搜索引擎”网站为例讲解如何设计合理的、适合网站特点的首页。

23.8.1　首页概述

设计网站首页要贴切网站主题、突出网站特色，并非复杂就可以。考虑“明日搜索引擎”网站主要用于为编程相关工作人员提供技术问题搜索和技术交流，借鉴互联网一些知名的搜索工具，设计出如图 23.20 所示的首页界面。“明日搜索引擎”网站的首页主要由网站 Logo、语言导航条、搜索区、友情链接和版权声明等部分组成，页面简洁大方、突出核心、便于用户使用。



图 23.20　明日搜索引擎网站首页

23.8.2　首页技术分析

“明日搜索引擎”网站首页，主要涉及两点较为关键的技术，分别为类似 Google 搜索条的制作和语言类别选项卡的制作，其中类似 Google 搜索条的制作过程已经在本堂课关键技术中进行了讲解，这里不再赘述，下面重点对语言选项卡的实现过程进行技术分析。

与 Visual Basic、C#等一些可视化编程语言相比，网页视图层的一些组件和特效的制作需要通过编码实现，而非简单的拖拽，这样制作出的组件更具灵活性。制作页面特效一般通过 JavaScript、Ajax 等技术实现。

制作“明日搜索引擎”网站语言类别选项卡的基本思想是：首先从数据库中提取所要显示语言类别的名称，并通过<div>标签显示在同一行，然后为这些<div>标签指定背景图片，最后定义如下方法（这里用 jQuery 实现）控制选项卡之间的切换。

代码位置：光盘\MR\23\application\modules\default\views\scripts\index\index.phtml

```
function changeSearchType(x, y)
{	//该方法中的两个参数 x,y 分别为每个选项卡按钮的 ID 和选项卡对应语言的 ID
	//更改全部选项卡背景图片
	$("#st_0").css("background", "url(<?php echo $this->baseUrl('/img/main_dh_btn2.gif')?>)");
	$("#st_0").css("color", "#333333");				//更改全部选项卡前景色
<?php foreach ($this->types as $type):?>
		$("#st_<?php echo $type['tid']?>").css("background", "url(<?php echo $this->baseUrl('/img/main_dh_btn2.gif')?>)");				//更改语言类别选项卡背景图片
		$("#st_<?php echo $type['tid']?>").css("color", "#333333");	//更改语言类别选项卡前景色
<?php endforeach;?>
	//更改当前选中的选项卡背景图片
	$("#"+x).css("background", "url(<?php echo $this->baseUrl('/img/main_dh_btn1.gif')?>)");
	$("#"+x).css("color", "#FFFFFF");		//更改当前选中的选项卡前景色
❶		$("#typeid").val(y);			//更改隐藏域 typeid 的值为当前选中选项卡对应的语言 ID
}
```

代码导读：

❶ jQuery 中的 val()方法，用来获取或设置指定元素的 value 值。

上述代码中，通过 jQuery 的 css()方法更改<div>标签的背景图片和前景色，该方法可以传入两个参数，第一个参数为要更改的 CSS 样式属性名称，如 color、background-color、font-size 等；第二个参数为属性所指定的值。其中第二个参数可以省略，如果设置第二个参数，则为 CSS 样式重新赋值，否则将获取该 CSS 样式的值。

23.8.3　首页的实现过程

“明日搜索引擎”网站采用 MVC 开发模式，所以其开发过程也需要按模型层、视图层和控制器层进行。在对首页进行技术分析时已经介绍到，制作该站首页需要从数据库中提取所有语言类别，所以应该在类别表的模型 Model_DbTable_Bbstype 中建立一个用来查询全部类别的方法，这里命名为 findAll()，其代码如下：

代码位置：光盘\MR\23\application\models\DbTable\Bbstype.php

```
public function findAll ($isCache = false) {
	$innerSelect1 = $this->getAdapter()->select();			//所有主题
	$innerSelect1->from('tb_title', 'count(*)')->join($this->_name, 'tb_title.bbstype_id=' . $this->_name . '.id', null)->where($this->_name . '.id = tid');
	$innerSelect2 = $this->getAdapter()->select();			//今日主题
```

```
    $innerSelect2->from('tb_title', 'count(*)')->join($this->_name, 'tb_title.bbstype_id=' . $this->_name . '.id',
null)->where('date(tb_title.addtime)=date(now())')->where($this->_name . '.id = tid');
    $innerSelect3 = $this->getAdapter()->select();            //最后主题 ID
    $innerSelect3->from('tb_title', 'tb_title.id')->join($this->_name, 'tb_title.bbstype_id=' . $this->_name . '.id',
null)->where($this->_name . '.id = tid')->order('tb_title.addtime desc')->limit(1, 0);
    $innerSelect4 = $this->getAdapter()->select();            //最后主题
    $innerSelect4->from('tb_title', 'tb_title.title')->join($this->_name, 'tb_title.bbstype_id=' . $this->_name . '.id',
null)->where($this->_name . '.id = tid')->order('tb_title.addtime desc')->limit(1, 0);
    $innerSelect5 = $this->getAdapter()->select();            //最后主题时间
    $innerSelect5->from('tb_title', 'tb_title.addtime')->join($this->_name, 'tb_title.bbstype_id=' . $this->_name .
'.id', null)->where($this->_name . '.id = tid')->order('tb_title.addtime desc')->limit(1, 0);
    $innerSelect6 = $this->getAdapter()->select();            //总回复
    $innerSelect6->from('tb_reply', 'count(*)')->join($this->_name, 'tb_reply.bbstype_id=' . $this->_name . '.id',
null)->where($this->_name . '.id = tid');
    $innserSelect7 = $this->getAdapter()->select();           //最后回复用户 ID
    $innserSelect7->from('tb_title', 'tb_title.user_id')->join($this->_name, 'tb_title.bbstype_id=' . $this->_name .
'.id', null)->where($this->_name . '.id=tid')->order('tb_title.addtime desc')->limit(1, 0);
    $innserSelect8 = $this->getAdapter()->select();            //最后回复用户昵称
    $innserSelect8->from('tb_user', 'netname')->where('id=(' . $innserSelect7 . ')');
    $select = $this->getAdapter()->select();                  //外层查询
    $select->from($this->_name, array($this->_name . '.id as tid' , $this->_name . '.typename' , $this->_name .
'.description' , $this->_name . '.addtime' , '(' . $innerSelect1 . ') as totaltitle' , '(' . $innerSelect2 . ') as
totaltodaytitle' , '(' . $innerSelect3 . ') as lasttitleid' , '(' . $innerSelect4 . ') as lasttitletitle' , '(' . $innerSelect5 . ') as
lasttitletime' , '(' . $innerSelect6 . ') as totalreply' , '(' . $innserSelect8 . ') as lasttitlenetname'))->order('' .
$this->_name . '.addtime asc');
    if ($isCache) {                                            //如果缓存查询结果
        if (! $result = $this->_cache->load(strtoupper($this->_name . '_findAll'))) {
//判断缓存文件是否存在或缓存是否过期
            $result = $this->getAdapter()->fetchAll($select);  //如果缓存文件不存在或缓存过期，则重写查找
$this->_cache->save($result, strtoupper($this->_name . '_findAll'));
//写入并生成新缓存
        }
    } else {
        $result = $this->getAdapter()->fetchAll($select);      //不缓存则直接查找
    }
    return $result;                                            //返回结果
}
```

上述代码中查询全部语言类别通过 Zend_Select 并使用子查询及 Zend_Cache 缓存技术实现。

首先介绍如何使用 Zend_Select 进行查询。使用 Zend_Select 进行查询，需要获得数据库适配器对象，然后通过该对象的 select()方法生成 Zend_Select 对象，代码如下：

```
$select = $this->getAdapter()->select();
```

其中，$this 代表当前表的模型，由于该模型继承自 Zend_Db_Table 类，所以通过$this->getAdapter()方法即可获取数据库适配器。获得 Zend_Select 对象后，就可以通过该对象的 from()方法，指定查询的表名、字段名；通过 order()方法指定记录排序方式。

那么如何通过 Zend_Select 进行子查询呢？其原理是：首先定义一个 Zend_Select 对象作为子查询，然后在外层查询中，将该 Zend_Select 对象作为查询结果的一个字段出现，代码如下：

```
$inSelect = $db->select();                                                    //内层查询，这里只给出关键代码
$outSelect = $db->select();                                                   //外层查询
$outSelect->from(表名, array('字段 1', '字段 2', '('.$inSelect.') as  字段 3' ));     //指定外层查询表名和字段名
```

最后介绍如何对查询结果进行缓存输出。Zend Framework 框架中，使用 Zend_Cache 对数据进行缓存来提高程序执行效率。在制作上述查询全部语言类别方法时，为了降低对数据库频繁查询的次数，降低数据库服务器负载，提交程序执行效率，这里使用 Zend_Cache 对查询结果进行缓存。Zend_Cache 通过其静态方法 factory()获得其实例，如下所示：

```
$cache = Zend_Cache::factory(string frontendName, string backendName, array frontendOptions, array backendOptions);
```

Zend_Cache 的 factory()方法的参数说明如表 23.7 所示。

表 23.7　Zend_Cache 的 factory()方法的参数说明

类　型	说　明
frontendName	必要参数，缓存前端名称
backendName	必要参数，缓存后端名称
frontendOptions	可选参数，缓存前端选项
backendOptions	可选参数，缓存后端选项

为了更加明确地了解 Zend_Cache 的使用方法，下面以本项目为例按步讲解 Zend Framework 中缓存的使用方法。

（1）为了便于以后维护，将缓存过期时间和缓存文件路径在 application.ini 配置文件中指定，如下所示：

```
#缓存时间
cache.leftTime = "3600"
#缓存文件存储目录
cache.cache_dir = APPLICATION_PATH "/tmp/"
```

（2）Zend_Cache 对象应该被项目全局应用，所以 Zend_Cache 对象在启动文件中被初始化。代码如下：

```
$config = $this->getOptions();//通过启动类的 getOptions()方法获取配置文件配置信息，并保存在多维数组中返回
Zend_Registry::set('config', $config);              //注册该多维数组
//缓存前端选项
$frontOptions = array('leftTime' => $config['cache']['leftTime'] , 'automatic_serialization' => true);
$backOptions = array('cache_dir' => $config['cache']['cache_dir']);                //缓存后端选项
$cache = Zend_Cache::factory('Core', 'File', $frontOptions, $backOptions);       //获取 Zend_Cache 对象
Zend_Registry::set('cache', $cache);                                              //注册 Zend_Cache 对象
```

（3）使用 Zend_Register 的静态方法 set()将缓存对象注册后，就可以在工程的其他模块中通过 Zend_Register 的 get()获取缓存对象并使用。代码如下：

```
private $_cache;                           //将缓存对象声明为私有成员，这样就可以在模型的其他方法中使用
public function init (){
    $this->_cache = Zend_Registry::get('cache');     //获取缓存对象
}
```

（4）在类别表模型的 findAll()方法中，通过缓存对象的 load()方法判断在缓存目录下是否有以该方法参数为缓存 ID 的缓存文件并且不过期，如果存在则从缓存文件中获取要查询的内容，否则从数据库中查询内容并保存在缓存文件中，如下所示：

```
if (! $result = $this->_cache->load(strtoupper($this->_name . '_findAll'))) { //判断缓存文件是否存在或过期
    $result = $this->getAdapter()->fetchAll($select);                        //如果缓存不存在或过期，重写查找
    $this->_cache->save($result, strtoupper($this->_name . '_findAll'));     //写入并生成新缓存
}
```

23.9 用户模块设计

用户作为网站活动的参与者和维护者，是网站能够生存发展的源动力。“明日搜索引擎”网站由于需要用户提出问题、解决问题，所以一个合理的用户模块显得十分重要。

23.9.1 用户模块概述

“明日搜索引擎”网站的用户模块，主要包括用户注册、登录、用户信息管理等功能。浏览者如果只进行问题查找，则无须注册，如果没有在该站点查找到预期的结果，那么可以将问题发布到“明日搜索引擎”网站中，其他用户就可以与之探讨、解答，这时就需要用户注册为本站会员。用户访问“明日搜索引擎”网站的流程如图 23.21 所示。

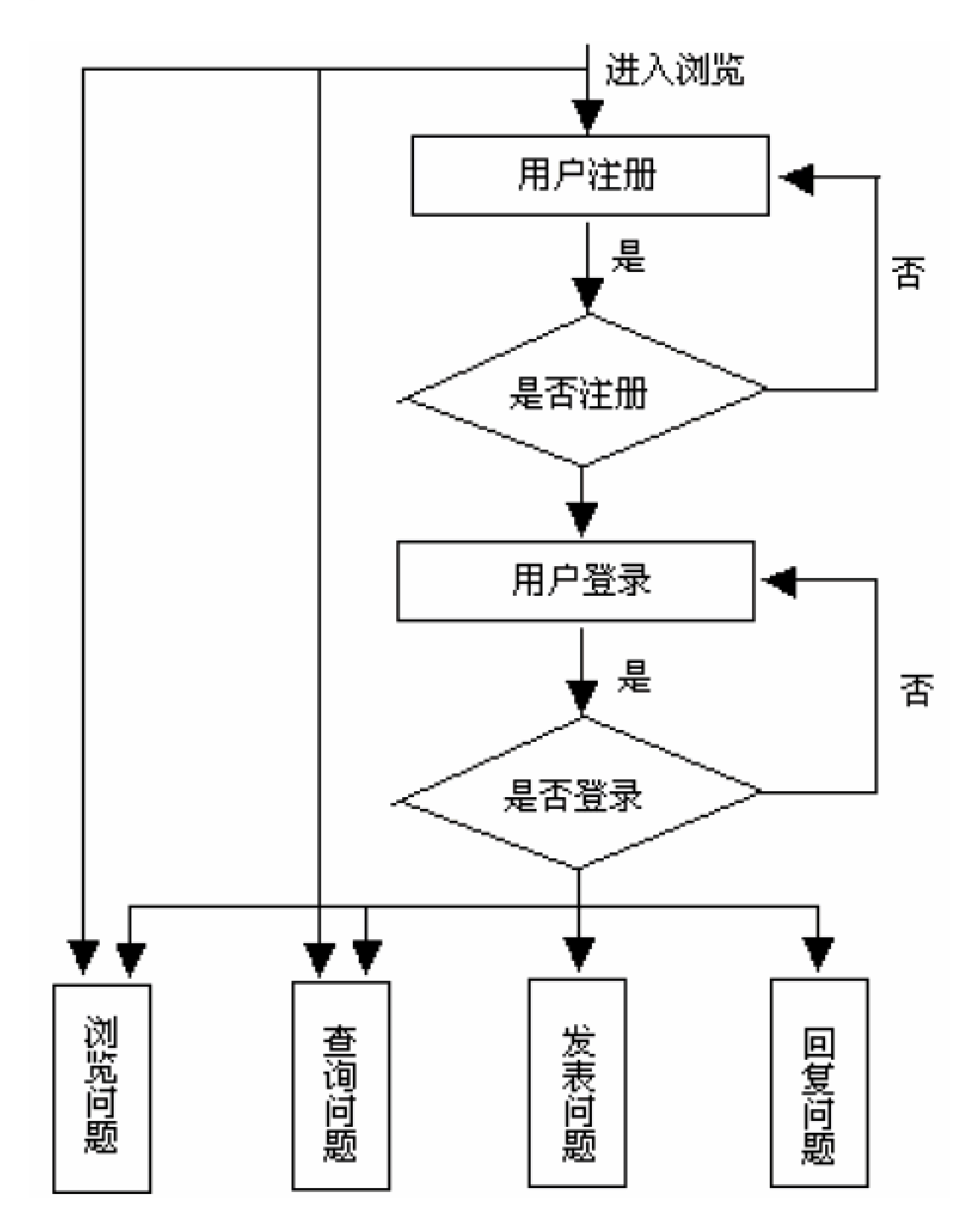


图 23.21 用户访问“明日搜索引擎”网站的流程

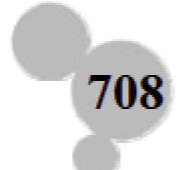
23.9.2 用户模块技术分析

为了防止非法用户通过恶意程序，采用试探密码的方式登录本站，或连续地向用户表中注入信息，“明日搜索引擎”网站的用户登录和注册表单中都增加了验证码，从而可以有效地预防上述情况发生。

在 Web 项目中，验证码可以有多种形式，如文字验证码、图片验证码、语言验证码等，本站所采用的是 Web 项目中最为常用的验证码形式，即图片验证码，如图 23.22 所示。



图 23.22 图片验证码



使用 PHP 语言制作验证码图片可以使用 GD2 函数绘制。考虑到验证码的高度、宽度以及图片中文字的字体在不同类型的表单中可能存在差异，在制作“明日搜索引擎”网站验证码时，单独将验证码的实现过程封装成验证码类。那么，验证码类是如何实现的呢？下面对此进行讲解。

（1）命名该类为 Plugin_Util_ValidateCode，这样命名的原因是如果将该验证码类保存在 application 目录下的 util 子目录中，并将该文件命名为 ValidateCode.php，那么在需要使用该验证码类时，根据 Zend Framework 的设计特点，不需要使用 include 等包含语句，就可以直接使用该验证码类。

定义该验证码类的数据成员，代码如下：

```
private $_width;                                    //宽度
private $_height;                                   //高度
private $_codeStr;                                  //验证码
private $_fontType;                                 //字体类型
private $_img;                                      //图像句柄
```

（2）定义验证码类的构造方法，实现验证码类的初始化，代码如下：

代码位置：光盘\MR\23\public_html\known\application\plugins\Util\ValidateCode.php

```
public function __construct ($width, $height, $codeStr = '0000', $fontType = 0) {
    $this->_width = $width;                         //验证码宽度
    $this->_height = $height;                       //验证码高度
    $this->_codeStr = substr($codeStr, 0, 4);       //验证码的值
    $this->_fontType = $fontType;                   //字体
}
```

（3）定义类的私有方法_getColor()，用于生成颜色句柄，代码如下：

代码位置：光盘\MR\23\public_html\known\application\plugins\Util\ValidateCode.php

```
private function _getColor ($x, $y){
    $r = mt_rand($x, $y);                           //红
    $g = mt_rand($x, $y);                           //绿
    $b = mt_rand($x, $y);                           //蓝
    return imagecolorallocate($this->_img, $r, $g, $b);   //返回颜色句柄
}
```

（4）定义私有方法_init()，用于创建图片验证码对象，代码如下：

代码位置：光盘\MR\23\public_html\known\application\plugins\Util\ValidateCode.php

```
private function _init (){
    $this->_img = imagecreate($this->_width, $this->_height);    //创建图像
}
```

（5）在创建的验证码图片对象的基础上创建验证码体，包括设置验证码的填充色、边框、验证码上的文字内容及字体等，其实现代码如下：

代码位置：光盘\MR\23\public_html\known\application\plugins\Util\ValidateCode.php

```
private function _build (){
imagefill($this->_img, 0, 0, $this->_getColor(150, 250));       //为图像填充背景色
//创建一个矩形作为验证码的边框
imagerectangle($this->_img, 0, 0, $this->_width - 1, $this->_height - 1, $this->_getColor(50, 150));
    if ($this->_fontType == 0) {                    //设置验证码文字的字体
        $fontFileName = 'ARIALBI.TTF';
    } else {
        $fontFileName = 'ARIALN.TTF';
    }
    for ($i = 0; $i < strlen($this->_codeStr); $i ++) {          //绘制文字
```

```
❶            imagettftext($this->_img, mt_rand(12, 24), 0, ($this->_width) / 4 * $i, mt_rand(20, $this->_height - 5), $this->_getColor(10, 180), APPLICATION_PATH . '/resources/' . $fontFileName, substr($this->_codeStr, $i, 1));
    }
    for ($i = 0; $i < 15; $i ++) {                                        //绘制 15 条干扰线
❷                imageline($this->_img, mt_rand(0, $this->_width), mt_rand(0, $this->_height), mt_rand(0, $this->_width), mt_rand(0, $this->_height), $this->_getColor(110, 210));
    }
}
```

代码导读：

❶ imagettftext()方法用于按指定的字体输出字符串。

❷ imageline()方法用于绘制直线。

（6）创建 show()方法用于显示图片，代码如下：

代码位置：光盘\MR\23\public_html\known\application\plugins\Util\ValidateCode.php

```
public function show (){
    header('content-type:image/png');                               //设置输出图片格式
    $this->_init();                                                 //图片初始化
    $this->_build();                                                //建立图片体
    imagepng($this->_img);                                          //输出图片
}
```

23.9.3 用户注册的实现过程

通过用户注册，可将用户的基本信息记录下来，便于用户参与站内活动及用户之间的交流。“明日搜索引擎”网站的注册表单如图 23.23 所示，该注册表单主要包括用户的昵称、登录密码、E-mail 地址、联系电话和所在地等信息。

给自己找个登录昵称：
小米
登录昵称为4-15个字符之间，用来区分注册用户，要求彼此不同
设置一个登录密码：
•••••••••
登录密码为6-15个字符之间
确认密码：
•••••••••
再输入一次密码，确保密码正确
E-mail：
mingrisoft@mingrisoft.com
请输入您的邮箱地址
联系电话：
12345678912
请输入您的联系电话，移动电话和固定电话均可
所在地：
吉林 请选择城市
请从列表中选择您目前所在地
验证码：
7219 看不清，我想换一张
请将将图片中的字符填写到验证码文本框，字母不区分大小写

图 23.23　用户注册表单

了解注册表单的结构后，下面讲解表单的实现过程。

（1）建立用户注册表单。“明日搜索引擎”网站的注册表单是通过 Zend Framework 的表单视图助手实现的，并通过 jQuery 技术对表单数据进行效验。注册表单 UI 实现代码请详见本书附带光盘。

（2）保存用户注册信息，并向注册用户的邮箱中发送注册成功提示。当用户填写完注册信息，并提交注册表单后，用户注册信息将被提交到用户控制器的注册动作中进行处理，代码如下：

代码位置：光盘\MR\23\application\modules\default\controllers\UserController.php

```
if ($this->_request->isPost()) {                          //如果用户通过 POST 方法提交表单
    $request = $this->_request;                           //获取当前控制器对象的 POST 方法
    $arrayProvice = Plugin_Util_ProvinceAndCityFactory::getProvince();    //获取用户所在省份
    $p = $arrayProvice[$request->getParam('p')];
    $arrayCity=Plugin_Util_ProvinceAndCityFactory::getCityByProvinceArrayIndex($request->getParam('p'));
    $c = $arrayCity[$request->getParam('c')];             //获取用户所在市

    $config = Zend_Registry::get('config');               //获取 Zend_Config 对象
    $arraySmtpConfig = array('auth' => 'login' , 'username' => $config['mail']['username'] , 'password' =>
$config['mail']['password']);                             //邮件配置信息
    $transport = new Zend_Mail_Transport_Smtp($config['mail']['host'], $arraySmtpConfig);    //邮件传输对象
❶   $mail = new Zend_Mail('utf-8');                       //构建 Zend_Mail 对象，并设置字符编码
    $mail->setSubject($config['mail']['subject']);        //邮件主题
    $mail->setBodyHtml(file_get_contents($config['mail']['bodyPath']));    //HTML 邮件内容
    $mail->setFrom($config['mail']['from'], $config['mail']['name']);      //发件人
    $mail->addTo(trim($request->getParam('email')));      //收件人
    try {
        $mail->send($transport);                          //发送邮件
    } catch (Zend_Exception $e) {
        $e->getMessage();
    }
    //保存用户信息到数据库
    $arrayUserInfo = array('netname' => trim($request->getParam('netname')) , 'password' => md5(trim($request->
getParam('password'))) , 'email' => trim($request->getParam('email')) , 'tel' => trim($request->getParam('tel')) ,
'pc' => $p . '-' . $c , 'regtime' => date('Y-m-d H:i:s') , 'face' => '' , 'pubtimes' => 0 , 'replytimes' => 0 , 'score' => 20 ,
'usertype' => 0);                                         //用户注册信息所构建的数组
    try {
        $this->_userModel->insert($arrayUserInfo);        //保存用户注册信息
    } catch (Zend_Exception $e) {
        $e->getMessage();
    }
    $this->_sessionNamespace->netname = trim($request->getParam('netname'));
    $this->_redirect('/user/register-success');           //定向到用户注册成功提示页面
    exit();
}
```

代码导读：

❶ 为 Zend_Mail 的构造方法指定 UTF-8 这个参数的作用是，设定邮件编码为 UTF-8 编码。

上述代码用于保存用户注册信息，主要是通过用户表模型 Model_DbTable_User 的 insert()方法实现的。insert()方法的语法如下：

```
insert(array insertArray)
```

该方法中主要包含一个关联数组参数，该数组的键名为用户表的字段名，键值为用户录入的注册信息。

为了提高网站形象，提高用户回访数量，在保存用户注册信息前，向注册用户所填写的邮箱中发送一封注册成功提示邮件。发送邮件功能是通过 Zend_Mail 实现的，使用 Zend_Mail 发送邮件相对简单，实例化 Zend_Mail 对象后，只需通过调用该类的几个方法，为邮件指定主题、内容、发件人和收件人等信息，最后再通过 send()方法将邮件发送出去。Zend_Mail 类中的常用方法如表 23.8 所示。

表 23.8　Zend_Mail 类中的常用方法

方　法	说　明
setSubject()	设置邮件主题
setBody()	设置文本格式的邮件内容
setBodyHtml()	设置 HTML 格式的邮件内容
setFrom()	设置发件人地址
addTo()	设置收件人地址

23.9.4　用户登录的实现过程

用户注册成功后，以后访问本站就可以直接以已注册的账号登录。“明日搜索引擎”网站的用户登录页面如图 23.24 所示。



图 23.24　用户登录页面

建立用户登录表单。“明日搜索引擎”网站的登录表单使用 Zend_Form 制作，代码如下：

代码位置：光盘\MR\23\application\forms\User\login.php

```
class Form_User_Login extends Zend_Form{                        //定义用户登录表单，继承自 Zend_Form
    public function __construct ($options = null) {
        parent::__construct($options);                          //调用父类构造函数
        $this->setName('form_login')                            //表单名称
             ->setMethod('post')                                //表单提交方法
             ->setAction($options['baseUrl'] . '/user/login')   //表单提交地址
             ->addAttribs(array(
                 'style' => 'margin:0px; padding:0px'           //表单样式
             ));
❶       $this->addElements(array(                               //为表单增加用户昵称录入文本框
             new Zend_Form_Element_Text('netname', array(
                 'required' => true,                            //指定该表单元素是必要的
```

```
        'label' => '昵称：',                                    //元素标签
        'attribs' => array(                                     //元素属性
            'class' => 'input_login_form',
            'style' => 'position:absolute; left:32px; top:0px; width:180px; height:18px; line-height:18px'
        ),
        'filters' => array('StringTrim'),                       //过滤掉首尾空格
        'validators' => array(
            array('NotEmpty', true, array('messages' => '・请输入登录昵称'))   //昵称非空验证
        ),
        'decorators' => array(                                  //元素装饰器
            'ViewHelper',
            array('HtmlTag', array('tag'=>'dd')),
            array('Label', array('tag'=>'dt'))
         )
    )),
    new Zend_Form_Element_Password('password', array(   //为表单增加用户密码录入文本框
        'required' => true,                                     //指定该表单元素是必要的
        'label' => '密码：',                                    //元素标签
        'attribs' => array(                                     //密码框属性
            'class' => 'input_login_form',
            'style' => 'position:absolute; left:32px; top:0px; width:180px; height:18px; line-height:18px;'
        ),
        'filters' => array('StringTrim'),                       //过滤掉首尾空格
        'validators' => array(                                  //登录密码非空验证
            array('NotEmpty', true, array('messages' => '・请输入登录密码'))
        ),
        'decorators' => array(                                  //元素装饰器
            'ViewHelper',
            array('HtmlTag', array('tag' => 'dd')),
            array('Label', array('tag' => 'dt'))
         )
    )),
    new Plugin_Form_Element_Vcode('vcode', array(               //验证码
        'required' => true,                                     //指定该表单元素是必要的
        'label' => '验证码：',                                  //元素标签
        'attribs' => array(                                     //验证码属性
            'textClass' => 'input_login_form',
            'textStyle' =>'position:absolute; left:32px; top:0px; width:60px; height:18px; line-height:18px; ',
            'imageStyle' =>'position:absolute; left:100px; top:0px;',
            'spanStyle' => 'position:absolute; left:42px; top:37px; height:18px',
            'aClass' => 'a4',
            'functionName' => 'changeValidateCode()'
        ),
        'filters' => array('StringTrim'),                       //过滤掉首尾空格
        'validators' => array(
            array('NotEmpty', true, array('messages' => '・请输入验证码')),    //验证码非空验证
            new Plugin_Validate_VcodeRight()                    //验证码正确性验证
        ),
        'decorators' => array(                                  //元素装饰器
```

```
                    'ViewHelper',
                    array('HtmlTag', array('tag' => 'dd', 'style' => 'height:55px;')),
                    array('Label', array('tag' => 'dt'))
                )
            )),
            new Zend_Form_Element_Image('submitImage', array(      //提交图片按钮
                'required' => false,                               //该元素值可为空
                'label' => '',
                'src' => $options['baseUrl'] . '/img/btn_login.gif',       //图片地址
                'attribs' => array(
                    'style' => 'position:absolute; left:30px; top:0px;'
                )
            ))
        ));
    }
}
```

代码导读：

❶ addElements()方法用于将表单元素对象添加到表单中。

使用 Zend_Form 制作表单，除可构建表单 UI 界面外，还可以同时实现用户录入内容的过滤、验证等操作，从而可以很大程度上提高程序开发效率。使用 Zend_Form 构建表单组件的构造方法一般有两个参数，分别为表单元素名称和指定该元素属性的关联数组，该数组常用键值说明如表 23.9 所示。

表 23.9　Zend_Form 表单元素常用键值说明

键　　值	说　　明
required	用于指定表单元素值是否是必要的
label	表单元素标签
attribs	表单元素属性
filters	对表单元素进行过滤
validators	对表单元素进行效验
decorators	对表单元素 UI 进行装饰

23.10　发表问题模块设计

当用户在“明日搜索引擎”网站搜索问题时，如果没有查找到预期结果，可以将要解决的问题发表到该站，其他用户就可以对问题进行解答或探讨。

23.10.1　发表问题模块概述

如果用户已经注册为本站会员，并成功登录本站，则可以通过“明日搜索引擎”网站的发表问题模块将问题发表出去，其他用户可以回答该问题，最终可以帮助提问用户获得问题的最佳答案。“明日搜索引擎”网站的发表问题表单主要包括问题主题、问题类别、问题的内容描述、附件、验证码等表单元素，如图 23.25 所示。



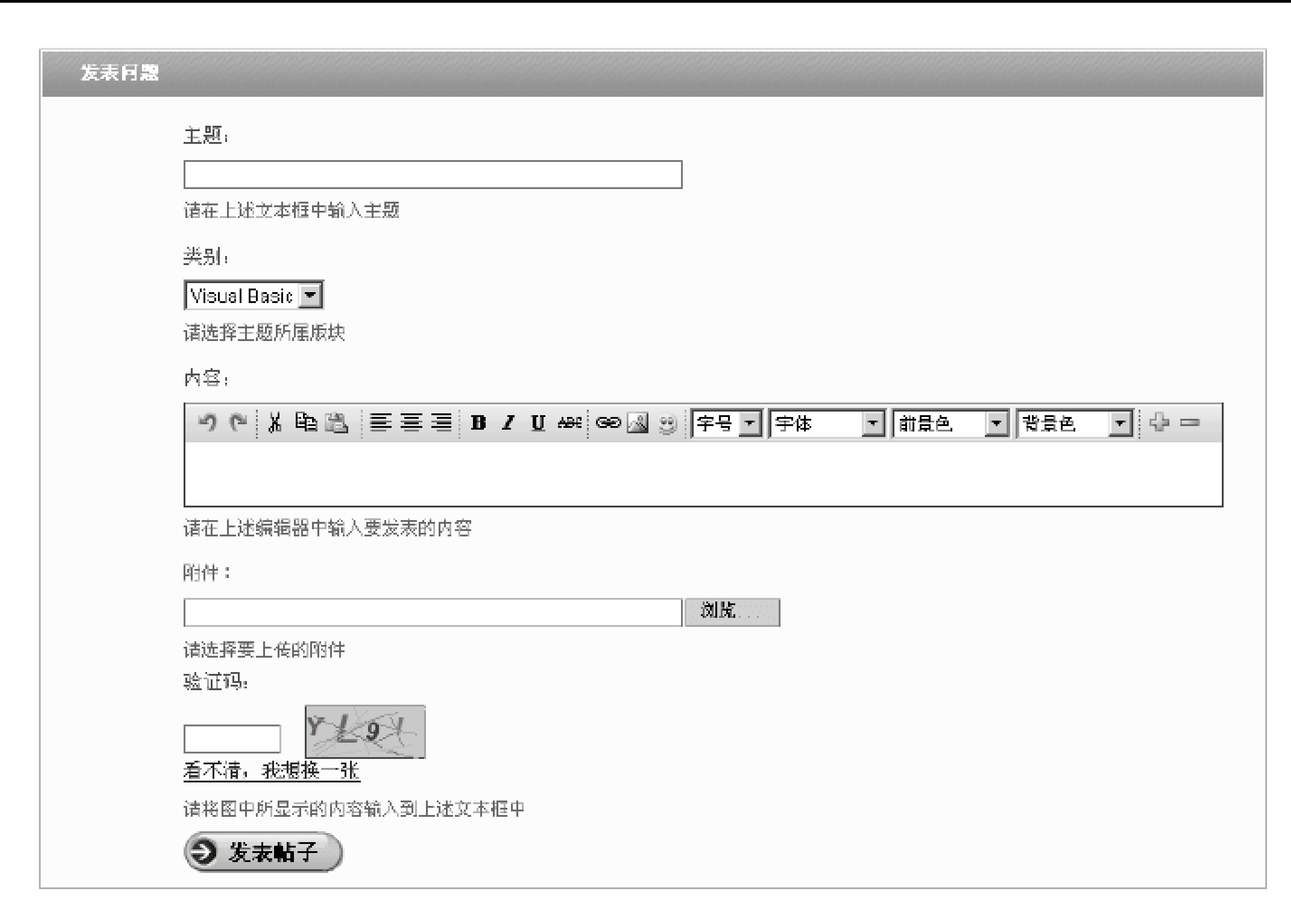


图 23.25　发表问题表单

23.10.2　发表问题模块技术分析

发表问题模块中，应该掌握的主要技术包括在线编辑器的制作、图片验证码的制作、文件上传的实现过程。其中，在线编辑器及图片验证码的制作过程已经在前面章节中进行了重点讲解，下面通过本模块介绍利用 Zend Framework 实现文件上传的过程。

（1）用户发表问题表单是通过 Zend_Form 实现的，所以需要使用 Zend_Form_Element_File 类建立文件上传域，代码如下：

代码位置：光盘\MR\23\application\forms\Bbs\Pubtitle.php

```
new Zend_Form_Element_File('file1', array(                    //文件选择域
    'required' => false,                                      //指定文件上传域并非必要
    'label' => '附件：',                                       //标签
    'maxFileSize' => '30000000',                              //上传文件大小
    'description' => '请选择要上传的附件',                      //描述
    'attribs' => array(                                       //文件上传域属性
        'size' => '60',
        'class' => 'input_pubtitle_form'
    ),
    'validators' => array(                                    //非空验证
        array('NotEmpty', false, array('messages' => '请输入内容'))
    ),
    'decorators' => array(                                    //装饰器
        'File',
        array('Description', array('tag' => 'dt')),
        array('HtmlTag', array('tag' => 'dd') ),
        array('Label', array('tag' => 'dt'))
    )
)),
```

在讲解用户登录表单时，已经详细介绍了 Zend_Form 表单及表单元素的常用属性的含义。上述代码中，使用'maxFileSize' => '30000000'指定所允许上传附带的最大值，运行发表问题模块，上述代码将转换为如下 HTML 代码：

```
<input type="hidden" name="MAX_FILE_SIZE" value="30000000" id="MAX_FILE_SIZE">
<input type="file" name="file1" id="file1" size="60" class="input_pubtitle_form">
```

（2）当用户提交发表问题表单后，用户提交的内容将被提交到 QuestionController 控制器的 pubtitle Action 动作进行处理，其中用户实现文件上传的关键代码如下：

代码位置：光盘\MR\23\application\modules\default\controllers\QuestionController.php

```
$adapter = new Zend_File_Transfer_Adapter_Http();       //构建上传适配器
$upfileName = '';                                        //上传后，保存在服务器中的文件名
if ($adapter->getFileName('file1') != null) {            //如果用户已经选择了上传文件
    $upfileDir = $this->_config['bbs']['upfilesdir'];    //从配置文件中读取上传文件保存目录
    if (! is_dir($upfileDir)) {                          //判断目录是否存在，如不存在则创建目录
        mkdir($upfileDir);
    }
    $adapter->setDestination($upfileDir);                //为上传适配器设置上传目录
$arrayOldFileName = array_reverse(explode('.', basename($adapter->getFileName('file1'))));
$extendsFileName = '.' . $arrayOldFileName[0];           //获取上传文件扩展名
//为上传文件指定新名称，用时将和一个 4 位随机数组成
    $upfileName = date('YmdHis') . mt_rand(1000, 9999) . $extendsFileName;
$upfilePathAndName = $upfileDir . '/' . $upfileName;     //文件在服务器中保存目录及名称
//更改上传文件名
    $adapter->addFilter('Rename', array('target' => $upfilePathAndName , 'overwrite' => true), 'file1');
    $adapter->addValidator('Size', false, array('min' => '0kB' , 'max' => '2MB' , 'bytestring' => false , 'messages'
=> '・您所上传的文件不能超过 2M'));                      //指定上传文件的范围
    $adapter->addValidator('ExcludeExtension', false, array('php' , 'exe' , 'messages' => '・您上传的文件类型不
允许'));                                                 //指定上传文件扩展名
}
```

上述代码实现文件上传是通过 Zend_File 实现的，由于本模块上传采用 HTTP 方式，所以实现文件上传主要通过 Zend_File_Transfer_Adapter_Http 适配器实现，该适配器的常用方法说明如表 23.10 所示。

表 23.10　Zend_File_Transfer_Adapter_Http 适配器常用方法说明

方　法	说　明
getFileName()	获取上传文件的名称
setDestination()	指定上传文件保存到服务器中的目录
addFilter()	为上传增加过滤规则
addValidator()	为上传增加效验规则

23.10.3　发表问题模块实现过程

制作“明日搜索引擎”网站发表问题模块，主要包括发表问题表单视图的建立和保存问题内容两个关键环节。

（1）“明日搜索引擎”网站发表问题表单使用 Zend_Form 实现，通过 Zend_Form 实现表单的制作可以简化对提交数据的过滤、效验、UI 布局的操作，从而能够提高开发效率，便于开发人员分层开发及日后生产环境下的维护工作。本站实现发表问题表单的代码如下：

代码位置：光盘\MR\23\application\forms\Bbs\Pubtitle.php

```
class Form_Bbs_Pubtitle extends Zend_Form{
    public function __construct ($options = null){
```



```
            parent::__construct(null);                                      //调用父类构造方法
            $this->setName('form_pubtitle')                                 //表单名称
                ->setMethod('post')                                         //表单提交方法
                ->setAction($options['baseUrl'].'/question/pubtitle')       //表单提交地址
                ->addAttribs(array(                                         //表单属性
                    'enctype' => 'multipart/form-data',
                    'style' => 'margin:0px; padding:0px;',
                ));
            $this->addElements(array(
                new Zend_Form_Element_Text('title', array(
                    ......                                  //省略代码用于设置问题主题元素的属性
                )),
                new Zend_Form_Element_Select('bbstypeid', array(
                    ......                                  //省略代码用于设置问题类别下拉列表属性
                )),
                new Plugin_Form_Element_Editor('content', array(
                    ......                                  //省略代码用于设置问题内容编辑器的属性
                 )),
                new Zend_Form_Element_File('file1', array(          //文件选择域
                    ......                                  //省略代码用于设置文件选择框的属性
                )),
                new Plugin_Form_Element_Vcode('vcode', array(
                    ......                                  //省略代码用于设置验证码的属性
                )),
                new Zend_Form_Element_Button('submitButton', array(
                    ......                                  //省略代码用于设置提交按钮的属性
                )),
                new Zend_Form_Element_Hidden('titleid', array(
 //在编辑问题时，该隐藏域用于保存问题 ID 的值
                    'value' => $options['titleid']
                ))
            ));
        }
}
```

上述 Form_Bbs_Pubtitle 类继承自 Zend_Form 类，这样该类就可以使用 Zend_Form 类的所有方法，如使用 setName()方法设置表单名称、setMethod()方法设置表单提交方法、setAction()方法设置表单提交地址等，其中 Zend_Form 表单元素常用类说明如表 23.11 所示。

表 23.11　Zend_Form 表单元素常用类说明

类　　名	说　　明
Zend_Form_Element_Text	生成文本框
Zend_Form_Element_Select	生成下拉列表框
Zend_Form_Element_File	生成文件选择域
Zend_Form_Element_Button	生成按钮
Zend_Form_Element_Hidden	生成隐藏域
Zend_Form_Element_Checkbox	生成复选框
Zend_Form_Element_Image	生成图片按钮
Zend_Form_Element_Password	生成密码框

续表

类　　名	说　　明
Zend_Form_Element_Radio	生成单选按钮
Zend_Form_Element_Reset	生成重置按钮
Zend_Form_Element_Submit	生成提交按钮
Zend_Form_Element_Textarea	生成文本域

（2）用户提交表单后，录入的内容将被提交到 QuestionController 控制器的 pubtitleAction 动作被保存，实现保存用户提交内容的核心代码如下：

代码位置：光盘\MR\23\application\modules\default\controllers\QuestionController.php

```
$pubUser = $this->_userModel->findByNetname($this->_auth->getIdentity());         //获取发帖用户信息
$arrayInsert = array('title' => $formData['title'] , 'content' => $formData['content'] , 'unhtmlcontent' => $this->view->
unHtml($formData['content']) ,
'addtime' => date('Y-m-d H:i:s') , 'istop' => false , 'user_id' => $pubUser['id'] , 'bbstype_id' => $formData['bbstypeid'] ,
'browse' => 0 , 'filename' =>
$upfileName , 'isjh' => false);                                              //发帖内容数组
$this->_titleModel->insert($arrayInsert);    //使用问题模型的 insert()方法，将问题信息保存到数据库
$lastTitle = $this->_titleModel->findLastByUserid($pubUser['id']);            //查找最后发帖信息
$this->_redirect('question/thread/param/' . $lastTitle['id']);                //使页面定位到该帖的详细信息页
```

上述代码首先通过用户模型的 findByNetname()方法获得当前登录用户的信息，然后将问题信息保存到数据库的 tb_title 表中，成功保存问题信息后，将页面定向到所发问题的详细信息页。

23.11　回复问题模块设计

用户成功注册为“明日搜索引擎”网站会员后，不仅可以发表问题，还可以回复问题，与其他用户进行讨论。

23.11.1　回复问题模块概述

“明日搜索引擎”网站的问题回复表单采用 Zend_Form 制作，主要包括回复主题、回复内容和验证码等表单元素，如图 23.26 所示。



图 23.26　问题回复表单

23.11.2　回复问题模块技术分析

回复问题模块中，应该掌握的主要技术包括在线编辑器的制作和使用方法、图片验证码的制作方法，以及问题引用的实现，其中在线编辑器和图片验证的实现方法在前面已经介绍过。下面介绍“明日搜索引擎”网站问题引用的实现过程，其关键代码如下：

```
$value = '<FIELDSET><LEGEND>引自：楼主</LEGEND>引用的内容</FIELDSET>';
```

将上述代码作为编辑器的 value 值，在编辑器中将显示如图 23.27 所示的效果，通过上述代码可知，引用可以通过 HTML 标签的<FIELDSET>标签实现，<FILELDSET>标签用于在文本或其他元素外绘制一个边框，如果在该标签内使用<LEGEND>子标签指定标题，则在边框左上侧还会显示指定的标题。

引自：楼主

我是今年刚工作的程序员，对一些代码的运用还不是很熟悉，对代码的书写还是不很规范。公司要我开发一个项目，我有点摸不着头绪，朋友买的编程词典我看了一下，觉得很实用，里面的代码是编辑好的，直接就可以使用，省了我不少的时间啊，经理还说我做得快，给我发了奖金，有机会一定要请你们吃饭呢

图 23.27　问题引用

23.11.3　回复问题模块实现过程

问题回复模块，主要包括回复表单的设计和保存回复信息两个主要过程。

（1）同样，“明日搜索引擎”网站的问题回复表单采用 Zend_Form 制作，其关键代码如下：

代码位置：光盘\MR\23\application\modules\default\controllers\QuestionController.php

```
class Form_Bbs_Reply extends Zend_Form{
    public function __construct ($options = null){
        parent::__construct(null);                  //调用父类构造函数
        $this->setName('form_reply')                //设置回复表单名称
             ->setMethod('post')                    //设置表单提交方法
//设置表单提交地址
->setAction($options['baseUrl'].'/question/thread/param/' . $options['titleid'] . '-1-T')
             ->addAttribs(array(                    //设置提交表单属性
                 'style' => 'margin:0px; padding:0px;',
        ));
        $this->addElements(array(
            new Zend_Form_Element_Text('title', array(
            ......                                  //省略代码用于设置回复主题文本框属性
            )),
            new Plugin_Form_Element_Editor('content', array(
            ......                                  //省略代码用于设置编辑器属性
            )),
            new Plugin_Form_Element_Vcode('vcode', array(
            ......                                  //省略代码用于设置验证码属性
            )),
            new Zend_Form_Element_Button('submitButton', array(
            ......                                  //省略代码用于设置提交按钮属性
            ))
        ));
    }
}
```

上述代码中的 Form_Bbs_Reply 类继承自 Zend_Form 类，通过 Zend_Form 类中定义的方法来为回复问题表单设置属性，并通过 addElements()方法将表单元素对象添加到问题回复表单中。

（2）用户输入完成回复内容，提交表单后，所有回复内容将被提交到 QuestionController 控制器的 threadAction 动作中进行保存，其中用于保存用户回复信息的关键代码如下：

代码位置：光盘\MR\23\application\modules\default\controllers\QuestionController.php

```
if ($this->_request->isPost()) {                                    //判断用户是否提交了表单
    $formData = $this->_request->getPost();                         //获取表单提交数据所组成的数组
    if ($replyForm->isValid($formData)) {                           //如果通过表单验证
        $replyid = $this->_request->getParam('replyid');            //获得问题类别 ID
        //如果设置了问题类别 ID，则说明要进行更改问题回复操作，否则进行添加问题回复操作
if (isset($replyid) && $replyid != null) {
$updateArray = array('title' => $formData['title'] , 'content' => $formData['content']);         //回复数组
            $where = $this->_replyModel->getAdapter()->quoteInto('id = ?', $replyid);       //更改条件
            $this->_replyModel->update($updateArray, $where);                                //更改数据
$this->_redirect('/question/thread/param/' . $titleid . '-' . $this->_request->getParam('topage') . '#r' . $replyid);
//重定向到问题详细信息页面
        } else {
            $replyUser = $this->_userModel->findByNetname($this->_auth->getIdentity());    //获取回复人
            $insertArray = array('title' => $formData['title'] , 'content' => $formData['content'] , 'addtime' =>
date('Y-m-d H:i:s') , 'user_id' => $replyUser['id'] , 'title_id' => $arrayTitle['tid'] , 'topindex' => 0 , 'bbstype_id' =>
$arrayTitle['bbstype_id']);                                         //添加问题回复数组
            $this->_replyModel->insert($insertArray);               //添加数据
            $totalReply = $arrayTitle['totalreply'] + 1;
❶           $totalPage = ceil($totalReply / $pageSize);
❷           $this->_redirect('/question/thread/param/' . $titleid . '-' . $totalPage . '#b');   //返回问题信息页面
        }
    } else {
        foreach ($replyForm->getMessages() as $messageArray) {      //获取错误信息
            foreach ($messageArray as $message) {
                $errorMsg .= $message . '<br />';
            }
        }
        $replyForm->populate($formData);                            //表单数据回填
    }
}
```

代码导读：

❶ 函数 ceil()的作用是对浮点数进行向上取整。

❷ _redirect()方法的作用是使当前页面重定向到其他页面。

上述代码首先通过当前控制器对象的_request 属性的 isPost()方法，判断用户是否提交了表单，如果是则首先对表单提交的数据进行数据效验，如果成功通过效验，则通过回复表（tb_reply）模型的 insert()方法，将回复信息保存到数据库中。

23.12　程序调试与错误处理

在程序开发过程中，难免会遇到各种错误，为了能够快速地解决程序开发过程中所遇到的问题，需要



开发人员具有良好的编程习惯和敏锐的观察能力，并不断地培养个人的开发经验，本节将对程序开发和运行过程中常见错误的调试和处理方法进行讲解。

23.12.1　项目开发过程中常见错误及解决方法

为了能够显示项目开发过程中的错误提示，需要开启 php.ini 文件的 display_startup_errors 和 display_errors 项，或者在本站的 application.ini 文件中加入如下配置来使项目在开发过程中显示错误提示。

```
phpSettings.display_startup_errors = 1
phpSettings.display_errors = 1
```

这里需要注意，在项目投入生产后，需要将上述两项的值设置为 0，这样在运行过程中即使出现错误，也不会在页面中打印出错误信息，从而可以防止网站的漏洞被非法用户发现，有效地提高网站的安全性。开启上述错误提示信息后，在项目开发过程中，一旦遇到错误，开发人员可以根据错误提示推断错误所在并予以解决。

在 PHP 5 以上版本中，引入了异常处理机制，开发人员通过异常处理可以捕获程序运行时的错误。异常处理中主要有 try、catch 和 throw 这 3 个关键字，其语法结构如下：

```
try{
    //要执行的代码块
}catch(Exception $e){
    //异常处理方式，这里可以打印错误信息或使用 throw 语句抛出异常
}
```

用户要执行的代码块编写在 try/catch 代码块之间，通过 catch 语句指定要捕获的异常，并通过 throw 语句抛出异常。

23.12.2　自定义错误页面

为了提高页面的友好程度，隐藏项目生产环境下的错误，经常会在项目中制作一个单独的用于显示错误提示的页面。“明日搜索引擎”网站的错误提示页面如图 23.28 所示。

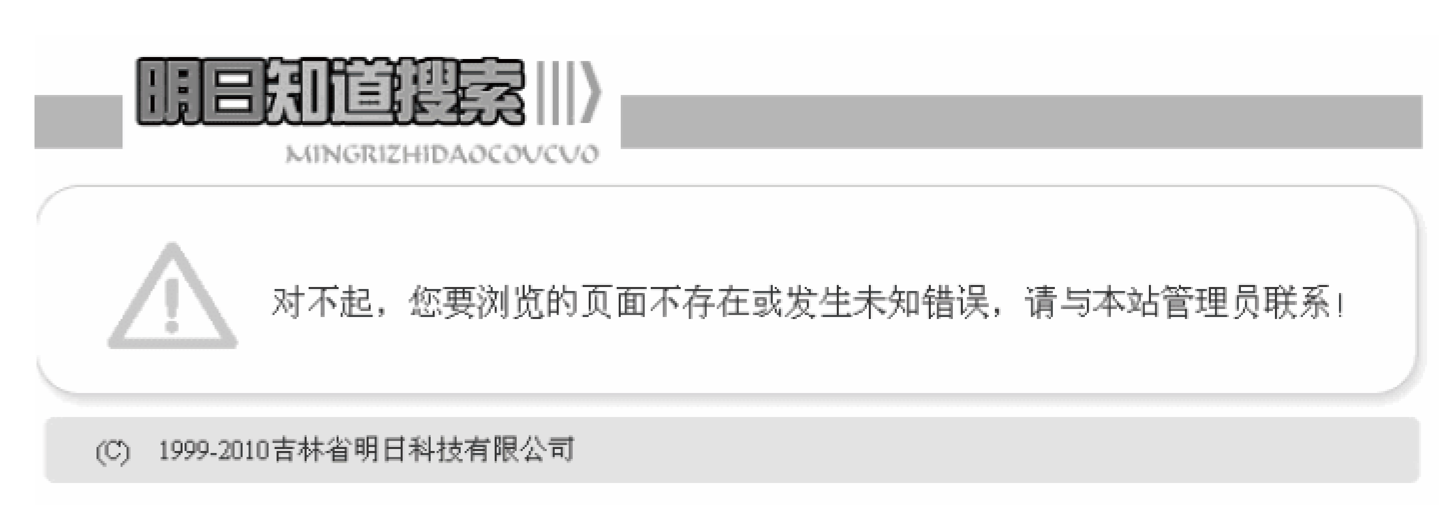


图 23.28　自定义的错误提示页面

使用 Zend Framework 框架制作项目时，有一个专门的错误控制器 ErrorController 用来显示页面错误提示，开发人员只需在该控制器中定义一个名为 errorAction 的动作即可。“明日搜索引擎”网站中的错误控制器代码如下：

代码位置：光盘\MR\23\application\modules\default\controllers\QuestionController.php

```
class ErrorController extends Zend_Controller_Action{
    public function errorAction (){
        $this->view->title = "明日搜索引擎网站错误提示";   //页面标题
```

```
            $this->_helper->layout->disableLayout();           //去掉布局
      }
}
```

定义完成上述错误控制器后，还需要定义 errro.phtml 视图，该视图用来显示错误提示的内容。

23.13 开 发 总 结

“明日搜索引擎”网站力求打造成为国内知名的专业软件开发技术搜索和交流平台，通过该平台开发人员和编程初学者可以查找到大部分开发和学习过程中可能会遇到的技术难题，即使没有查找到预期结果，还可以通过该平台将问题发表出去，这样就可以与其他网友探讨，最终将问题解决。

在开发过程中，通过严谨的需求分析、功能分析和数据库分析的全过程，并充分考虑到生产环境下的升级与维护等后期工作，本站采用 Zend 公司研发的 MVC 框架 Zend Framework 进行开发。在开发过程中，使用到目前 Web 开发领域的热门技术，如在线编辑器、图片验证码、类似 Google 搜索条的下拉关键字提示列表、关键字分词查询和查询结果中关键字描红输出等。

通过学习本堂课内容，可以对项目的完整开发流程有所了解，并理解 MVC 开发模式的含义。技术方面，不仅可以掌握目前 Web 应用的主流模块和功能的制作方法，而且可以熟练应用 Zend Framework 框架进行项目开发。如果能够潜心学习本堂课内容，相信定能受益匪浅，使个人技术水平有显著的提高。

第24堂课

学校图书馆管理系统

（视频讲解：150分钟）

随着网络技术的高速发展和计算机应用的普及，利用计算机对图书馆的日常工作进行管理势在必行。虽然目前很多大型的图书馆已经有一整套比较完善的管理系统，但是在一些中小型的图书馆中，大部分工作仍需由手工完成，工作起来效率比较低，管理员不能及时了解图书馆内各类图书的借阅情况，读者需要的图书难以在短时间内找到，不便于动态及时地调整图书结构。为了更好地适应当前读者的借阅需求，解决手工管理中存在的许多弊端，越来越多的中小型图书馆正在逐步向计算机信息化管理转变。本堂课通过开发一个流行的图书馆管理系统，为读者讲解详细的项目开发流程。

学习摘要：

- 图书馆管理系统开发的基本过程
- 系统设计的方法
- 如何分析并设计数据库、数据表
- 多表查询的方法
- 面向对象的编程方法
- 主要功能模块的实现方法
- 单元测试的方法
- 如何自动计算图书归还日期
- 内联接和外联接语句的使用方法

24.1　开 发 背 景

博考图书馆是一家吉林××师范学校的大型图书馆。随着学校图书馆规模的不断壮大，经营的图书品种、数量也逐渐增多。在学校图书馆不断发展的同时，常年采用的传统的人工方式管理暴露了一些问题。例如，查找读者借阅的某一本图书的具体摆放位置，需要靠人工记忆在书海中苦苦查找，由于图书储存量大，很难准确定位图书的具体位置，因此每天都要浪费大量宝贵的时间资源。学校图书馆为了提高工作效率，同时摆脱图书管理人员在工作中出现的种种弊端，现需要委托某单位开发一个学校图书馆管理系统。

24.2　需 求 分 析

通过计算机对图书进行管理，不仅为图书馆的管理注入了新的生机，而且在运营过程中节省了大量的人力、物力、财力和时间，可以提高图书馆的效率，还为图书馆在读者群中树立了一个全新的形象，并为图书馆日后发展奠定一个良好的基础。通过对一些大型图书馆的实际考察、分析，并结合图书馆的要求以及实际的市场调查，要求本系统具有以下功能：

- ☑ 网站设计页面要求美观大方、个性化，功能全面，操作简单。
- ☑ 要求实现基础信息的管理平台。
- ☑ 要求对所有读者进行管理。
- ☑ 要求实现图书借阅排行、了解当前的畅销书。
- ☑ 商品分类详尽，可按不同类别查看图书信息。
- ☑ 提供快速的图书信息、图书借阅检索功能，保证数据查询的灵活性。
- ☑ 实现图书借阅、图书续借、图书归还的功能。
- ☑ 实现综合条件查询，如按用户指定条件查询、按日期时间段查询、综合条件查询等。
- ☑ 要求图书借阅、续借、归还时记下每一笔记录的操作员。
- ☑ 实现对图书借阅、续借和归还过程的全程数据信息跟踪。
- ☑ 提供借阅到期提醒功能，使管理者可以及时了解到已经到达归还日期的图书借阅信息。
- ☑ 提供灵活、方便的权限设置功能，使整个系统的管理分工明确。
- ☑ 具有易维护性和易操作性。

24.3　系 统 设 计

24.3.1　系统目标

根据前面所做的需求分析及用户的需求可以得出，学校图书馆管理系统实施后，应达到以下目标：

- ☑ 网站设计页面要求美观大方、功能全面，操作简单。
- ☑ 网站整体结构和操作流程合理顺畅，实现人性化设计。
- ☑ 规范、完善的基础信息设置。
- ☑ 对操作员设置不同的操作权限，为管理员提供修改权限功能。



- ☑ 对所有读者进行集中管理。
- ☑ 对图书信息进行集中管理。
- ☑ 实现图书借阅排行，以便了解当前的畅销书。
- ☑ 提供快速的图书信息、图书借阅检索功能。
- ☑ 实现图书借阅、图书续借、图书归还功能。
- ☑ 实现综合条件查询，如按用户指定条件查询、按日期时间段查询、综合条件查询等。
- ☑ 实现图书借阅、续借、归还时记下每一笔记录的操作员。
- ☑ 支持图书到期提醒功能。
- ☑ 为操作员提供密码修改功能。
- ☑ 系统运行稳定、安全可靠。

24.3.2　系统功能结构

根据学校图书馆管理系统的特点，可以将其分为系统设置、读者管理、图书档案管理、图书借还和系统查询 5 个部分，其中各个部分及其包括的具体功能模块如图 24.1 所示。

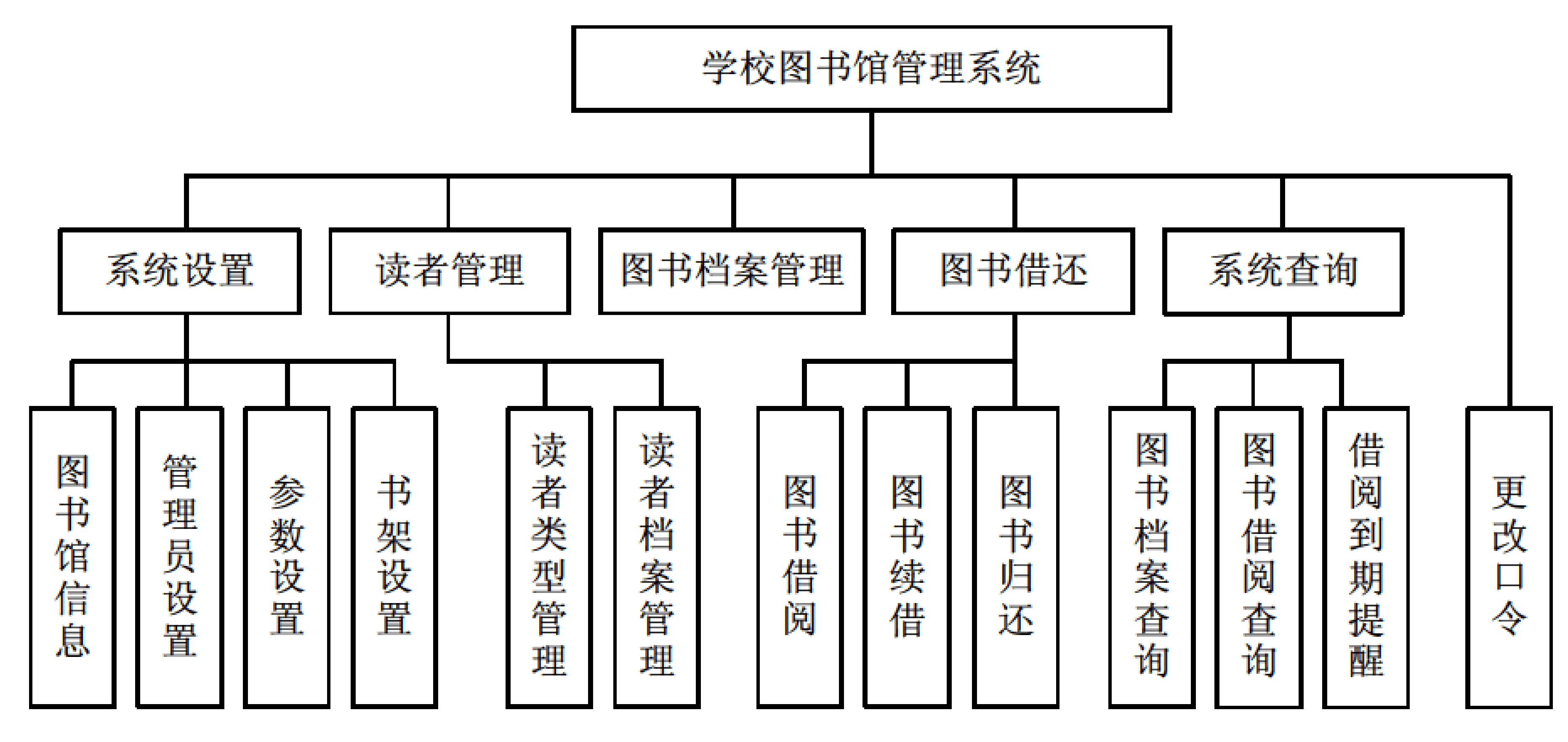


图 24.1　学校图书馆管理系统功能结构图

24.3.3　系统流程图

学校图书馆管理系统的流程如图 24.2 所示。

24.3.4　开发环境

在开发学校图书馆管理系统时，该项目使用的软件开发环境介绍如下。

1. 服务器端

- ☑ 操作系统：Windows Server 2003/Linux（推荐）。
- ☑ 服务器：Apache 2.2。
- ☑ PHP 软件：PHP 5.1.6。
- ☑ 数据库：MySQL 5.0.24。

☑ MySQL 图形化管理软件：phpMyAdmin-2.9.0.2。
☑ 开发工具：Dreamweaver 8。
☑ 浏览器：IE 6.0 及以上版本。
☑ 分辨率：最佳效果 1024 像素×768 像素。

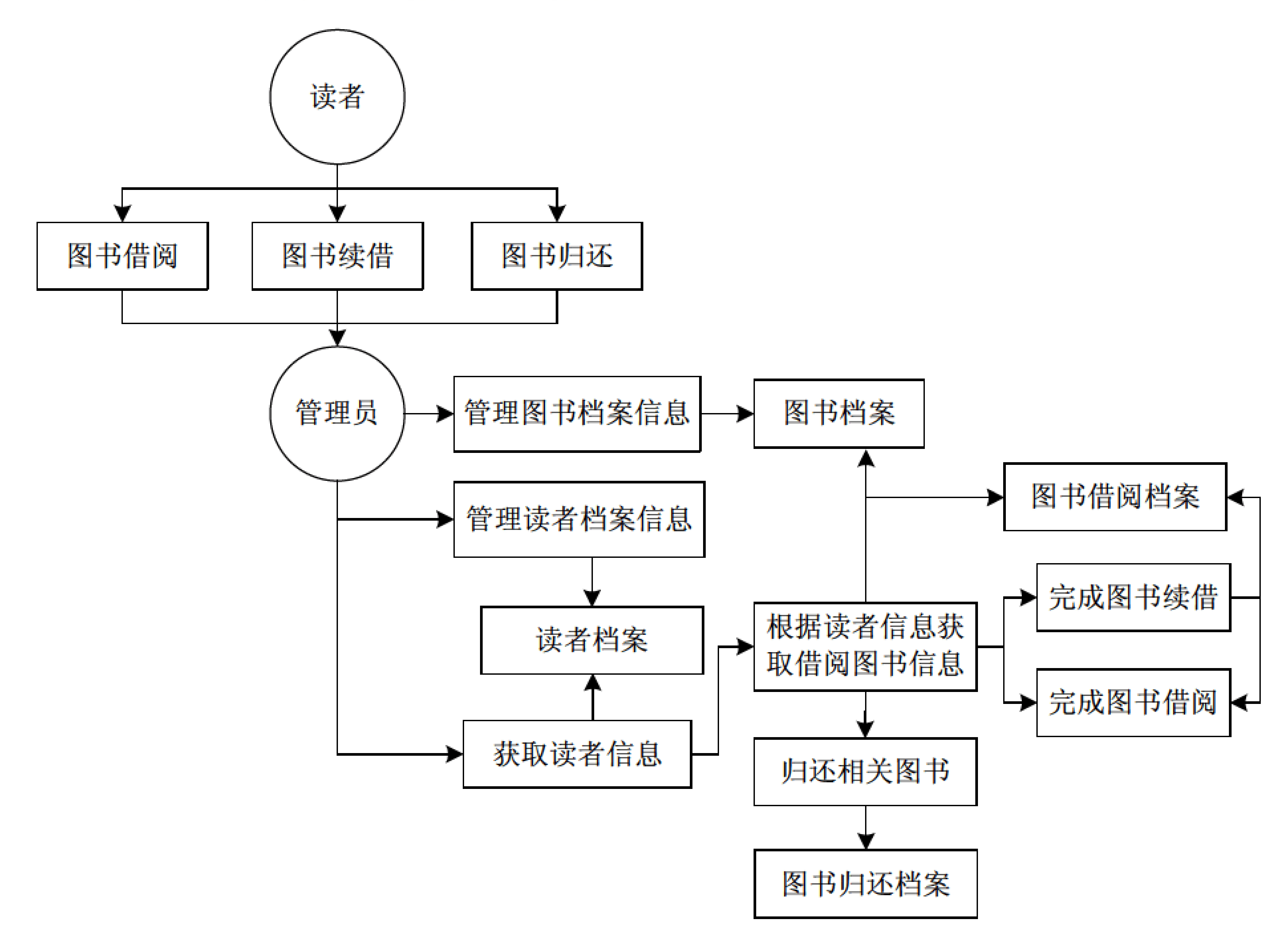


图 24.2　学校图书馆管理系统流程图

2．客户端

☑ 浏览器：IE 6.0 及以上版本。
☑ 分辨率：最佳效果 1024 像素×768 像素。

24.3.5　文件夹组织结构

在编写代码之前，可以把系统中可能用到的文件夹先创建出来（例如，创建一个名为 Images 的文件夹，用于保存网站中所使用的图片），这样不但可以方便以后的开发工作，也可以规范网站的整体架构。笔者在开发学校图书馆管理系统时，设计了如图 24.3 所示的文件夹组织结构图。在开发时，只需要将所创建的文件保存在相应的文件夹中即可。

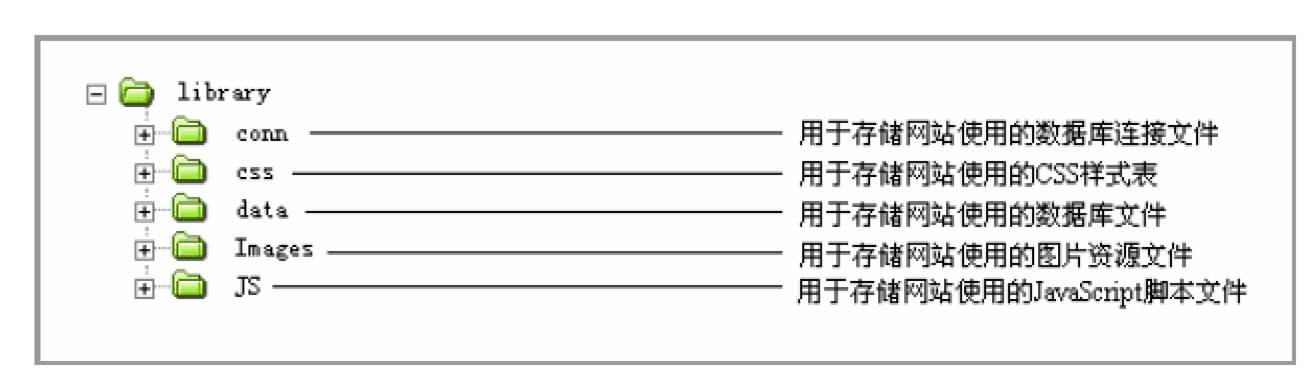


图 24.3　文件夹组织结构

24.4　数据库设计

学校图书馆管理系统是一个数据库开发的 Web 网站。下面对学校图书馆使用的数据库进行分析和介绍。

24.4.1　数据库分析

由于本系统是为中小型的图书馆开发的程序，需要充分考虑到成本问题及使用需求（如跨平台）等问题，而 MySQL 是世界上最为流行的开放源码的数据库，是完全网络化的跨平台的关系型数据库系统，这正好满足了中小型企业的需求，所以本系统采用 MySQL 数据库。

24.4.2　数据库概念设计

根据以上各节对系统所做的需求分析、系统设计，规划出本系统中使用的数据库实体分别为图书档案实体、读者档案实体、借阅档案实体、归还档案实体和管理员实体。下面将介绍几个关键实体的 E-R 图。

1．图书档案实体

图书档案实体包括编号、条形码、书名、类型、作者、译者、出版社、价格、页码、书架、录入时间和操作员等属性。图书档案实体的 E-R 图如图 24.4 所示。

2．读者档案实体

读者档案实体包括编号、姓名、性别、条形码、职业、出生日期、有效证件、证件号码、电话、电子邮件、登记日期、操作员、类型和备注等属性。读者档案实体的 E-R 图如图 24.5 所示。

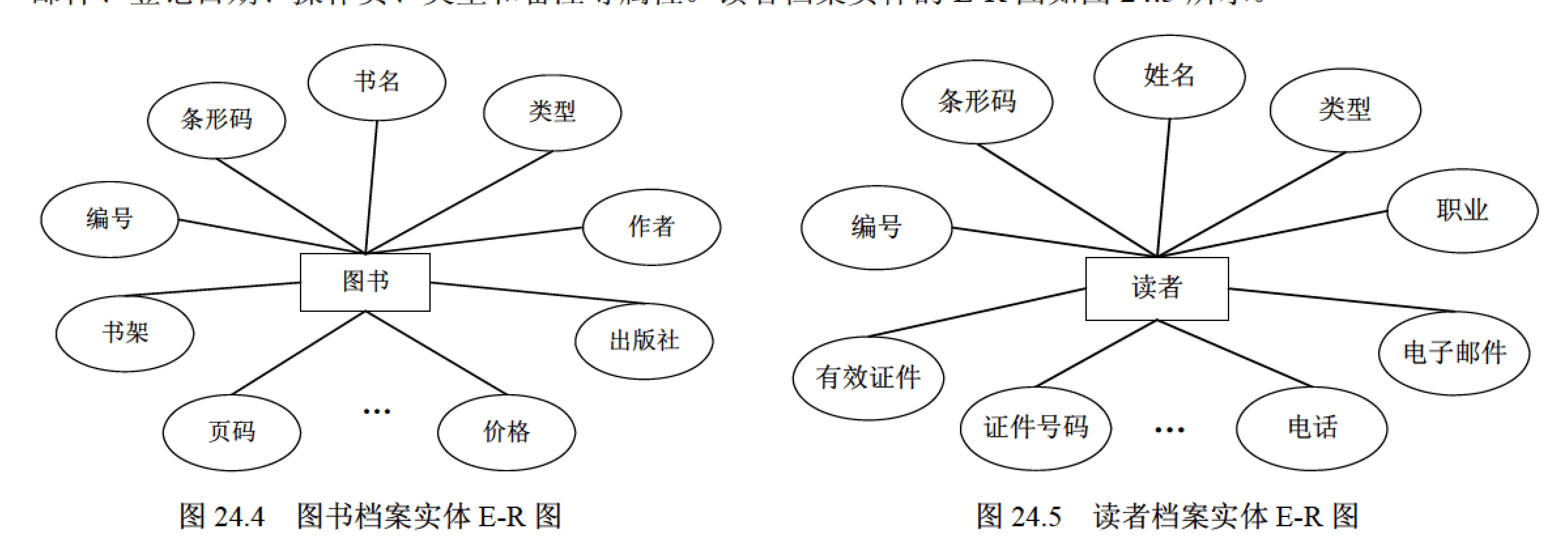


图 24.4　图书档案实体 E-R 图　　图 24.5　读者档案实体 E-R 图

3．借阅档案实体

借阅档案实体包括编号、读者编号、图书编号、借书时间、应还时间、操作员和是否归还等属性。借阅档案实体的 E-R 图如图 24.6 所示。

4．归还档案实体

归还档案实体包括编号、读者编号、图书编号、归还时间和操作员等属性。归还档案实体的 E-R 图如图 24.7 所示。

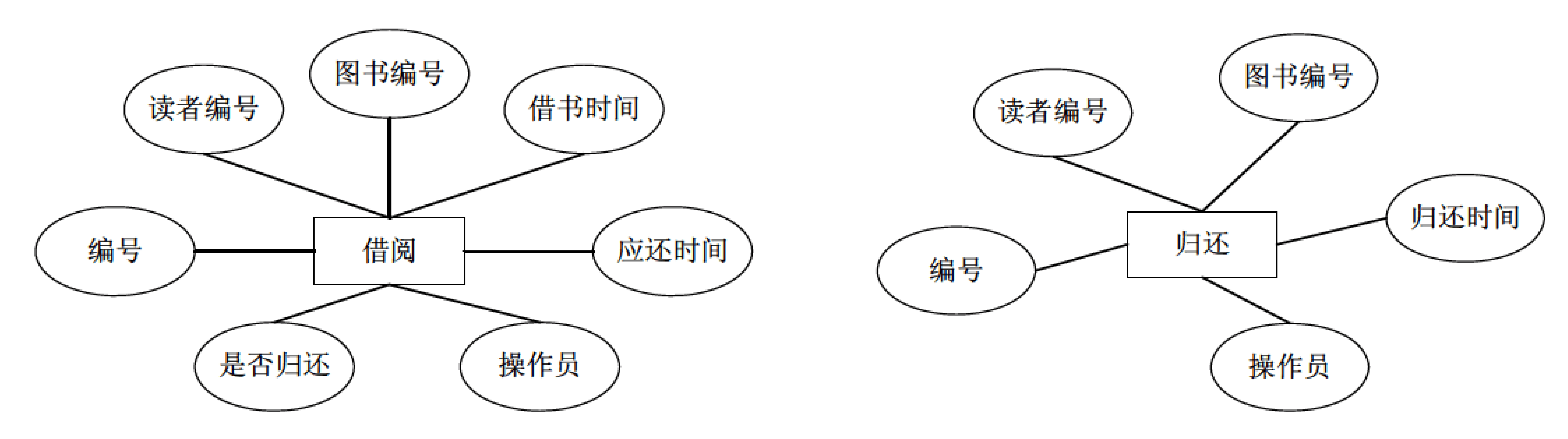


图 24.6　借阅档案实体 E-R 图　　　　图 24.7　归还档案实体 E-R 图

24.4.3　创建数据库及数据表

结合实际情况及对用户需求的分析，学校图书馆管理系统 db_database24 数据库主要包含如下 9 个数据表，如图 24.8 所示。

结合该数据表的创建方法，读者可以自行创建以下数据表。数据表的设计结构如图 24.9～图 24.11 所示。

1．tb_bookinfo（图书信息表）

图书信息表主要用于存储图书的基础信息。该数据表的结构如图 24.9 所示。

服务器: localhost ▸ 数据库: db_database24

表	类型	整理	说明
tb_bookcase	MyISAM	gb2312_chinese_ci	图书书架信息表
tb_bookinfo	MyISAM	gb2312_chinese_ci	图书信息表
tb_borrow	MyISAM	gb2312_chinese_ci	图书借阅信息表
tb_library	MyISAM	gb2312_chinese_ci	图书馆信息表
tb_manager	MyISAM	gb2312_chinese_ci	管理员信息表
tb_parameter	MyISAM	gb2312_chinese_ci	参数设置信息表
tb_publishing	MyISAM	gb2312_chinese_ci	出版社信息表
tb_purview	MyISAM	gb2312_chinese_ci	权限信息表
tb_reader	MyISAM	gb2312_chinese_ci	读者信息表
tb_readertype	MyISAM	gb2312_chinese_ci	读者类型信息表

图 24.8　学校图书馆管理系统数据表



服务器: localhost ▸ 数据库: db_database24 ▸ 表 : tb_bookinfo

字段	类型	整理	Null	默认	额外	说明
barcode	varchar(30)	gb2312_chinese_ci	否			图书条形码
bookname	varchar(70)	gb2312_chinese_ci	是	NULL		图书名称
typeid	int(10)		是	NULL		图书类型
author	varchar(30)	gb2312_chinese_ci	是	NULL		图书作者
translator	varchar(30)	gb2312_chinese_ci	是	NULL		图书译者
ISBN	varchar(20)	gb2312_chinese_ci	是	NULL		图书ISBN
price	float(8,2)		是	NULL		图书定价
page	int(10)		是	NULL		图书页码
bookcase	int(10)		是	NULL		图书书架
storage	int(10)		是	NULL		图书库存
inTime	date		是	NULL		入库时间
operator	varchar(30)	gb2312_chinese_ci	是	NULL		操作员
del	tinyint(1)		是	NULL		是否删除
id	int(11)		是	NULL	auto_increment	自动编号id

图 24.9　图书信息表结构

2．tb_borrow（图书借阅信息表）

图书借阅信息表主要用于存储图书的借阅信息。该数据表的结构如图 24.10 所示。

服务器: localhost ▸ 数据库: db_database24 ▸ 表 : tb_borrow

字段	类型	整理	Null	默认	额外	说明
id	int(10)		否		auto_increment	自动编号id
readerid	int(10)		是	NULL		读者id编号
bookid	int(10)		是	NULL		图书id编号
borrowTime	date		是	NULL		图书借阅时间
backTime	date		是	NULL		图书归还时间
operator	varchar(30)	gb2312_chinese_ci	是	NULL		操作员
ifback	tinyint(1)		是	0		是否归还

图 24.10　图书借阅信息表结构



3．tb_reader（读者信息表）

读者信息表主要用于存储用户的读者的基础信息。该数据表的结构如图 24.11 所示。

服务器: localhost ▸ 数据库: db_database24 ▸ 表 : tb_reader

字段	类型	整理	Null	默认	额外	说明
id	int(10)		否		auto_increment	自动编号id
name	varchar(20)	gb2312_chinese_ci	是	NULL		读者姓名
sex	varchar(4)	gb2312_chinese_ci	是	NULL		性别
barcode	varchar(30)	gb2312_chinese_ci	是	NULL		读者条形码
vocation	varchar(50)	gb2312_chinese_ci	是	NULL		读者职业
birthday	date		是	NULL		出生日期
paperType	varchar(10)	gb2312_chinese_ci	是	NULL		读者证件
paperNO	varchar(20)	gb2312_chinese_ci	是	NULL		证件号码
tel	varchar(20)	gb2312_chinese_ci	是	NULL		读者电话
email	varchar(100)	gb2312_chinese_ci	是	NULL		E-mail地址
createDate	date		是	NULL		办卡时间
operator	varchar(30)	gb2312_chinese_ci	是	NULL		操作员
remark	mediumtext	gb2312_chinese_ci	是	NULL		备注
typeid	int(11)		是	NULL		读者类型

图 24.11　读者信息表结构

说明：限于篇幅，笔者在此只给出较重要的数据表，其他数据表参见本书附带的光盘。

24.5　首 页 设 计

24.5.1　首页概述

管理员通过“系统登录”模块的验证后，可以登录到图书馆管理系统的首页。系统首页主要包括导航栏、排行榜和版权信息 3 部分。其中，导航栏中的功能菜单将根据登录管理员的权限进行显示。例如，系统管理员 Tsoft 登录后，将拥有整个系统的全部功能，因为它是超级管理员。

下面是本案例中提供的系统首页，该页面在本书光盘中的路径为\mr\24\index.php，如图 24.12 所示。



图 24.12　学校图书馆管理系统首页

24.5.2 首页技术分析

学校图书馆管理系统是一个功能全面、大型的 Web 网站，通过对网站的安全性考虑，本网站对该系统进行权限的分配，只有管理员级别的超级用户可以对普通用户的权限进行管理和设置。系统首页主要通过判断管理员的权限来显示该用户所操作的功能模块，关键代码如下：

代码位置：光盘\mr\24\navigation.php

```
<?php
session_start();                                                   //初始化 session 变量
include("conn/conn.php");                                          //连接数据库文件
$query=mysql_query("select  m.id,m.name,p.id,p.sysset,p.readerset,p.bookset,p.borrowback,p.sysquery  from
tb_manager as m left join (select * from tb_purview ) as p on m.id=p.id where name='$_SESSION[admin_
name]'");
$info=mysql_fetch_array($query);                                   //检索用户权限
?>
<!--检索用户所对应的权限，如果权限值为 1，则说明该功能可用，并输出到浏览器，否则不显示-->
<td width="70%" align="right">
<a href="index.php" class="a1">首页</a>  ¦
 <?php if($info[sysset]==1){ ?><a  onmouseover=showmenu(event,sysmenu) onmouseout=delayhidemenu()
style="CURSOR:hand" class="a1">系统设置</a>  ¦  <?php } ?>
<?php if($info[readerset]==1){?><a  onmouseover=showmenu(event,readermenu) onmouseout=delayhidemenu()
style= "CURSOR:hand" class="a1">读者管理</a>  ¦  <?php } ?>
<?php if($info[bookset]==1){ ?><a href="book.php" class="a1">图书档案管理</a>  ¦  <?php }?>
<?php if($info[borrowback]==1){?><a  onmouseover=showmenu(event,borrowmenu) onmouseout=delayhidemenu()
style="CURSOR:hand"class="a1" >图书借还</a>  ¦  <?php }?>
<?php if($info[sysquery]==1){ ?><a  onmouseover=showmenu(event,querymenu) onmouseout=delayhidemenu()
style="CURSOR:hand" class="a1">系统查询</a>  ¦  <?php } ?>
<a   href="pwd_Modify.php" class="a1">更改口令</a>  ¦
<a href="safequit.php" class="a1">注销</a>
</td>
```

说明：在权限信息表 tb_purview 中，权限值为 1，代表具备该模块的操作权限；权限值为 0，代表不具备该模块的操作权限。

在实现系统导航菜单时，引用了 JavaScript 文件 menu.JS，该文件中包含全部实现半透明背景菜单的 JavaScript 代码。

技巧：将页面中所涉及的 JavaScript 代码保存在一个单独的 JS 文件中，然后通过<script></script>将其引用到需要的页面，可以规范页面代码。在系统导航页面引用 menu.JS 文件的代码如下：

```
<script src="JS/menu.JS"></script>
```

24.5.3 首页的实现过程

系统首页的内容显示区用于显示图书的排行信息，并将排行结果按借阅数量降序排列。该页的关键代码如下：

代码位置：光盘\mr\24\index.php

```
<?php
include("conn/conn.php");                                          //连接数据源文件
```



```
$sql=mysql_query("select * from (select bookid,count(bookid) as degree from tb_borrow group by bookid) as borr join (select b.*,c.name as bookcasename,p.pubname,t.typename from tb_bookinfo b left join tb_bookcase c on b.bookcase=c.id join tb_publishing p on b.ISBN=p.ISBN join tb_booktype t on b.typeid=t.id where b.del=0) as book on borr.bookid=book.id order by borr.degree desc limit 10");
$info=mysql_fetch_array($sql);                                   //检索图书借阅信息
$i=1;
do{                                                              //应用 do…while 循环语句显示图书信息
?>
<tr>
     <td height="25" align="center"><?php echo $i;?></td>
     <td style="padding:5px;"> <?php echo $info[barcode];?></td>
     <td style="padding:5px;"><?php echo $info[bookname];?></td>
     <td style="padding:5px;"><?php echo $info[typename];?></td>
     <td align="center"> <?php echo $info[bookcasename];?></td>
     <td align="center"> <?php echo $info[pubname];?></td>
     <td align="center"><?php echo $info[author];?></td>
     <td align="center"><?php echo $info[price];?></td>
     <td align="center"><?php echo $info[degree];?></td>
</tr>
<?php
 $i=$i+1;                                                        //变量自加 1 操作
 }while($info=mysql_fetch_array($sql));                          //do…while 循环语句结束
?>
```

24.6　管理员模块设计

24.6.1　管理员模块概述

管理员模块主要包括管理员登录、查看管理员列表、添加管理员信息、管理员权限设置、管理员删除和更改口令 6 个功能。管理员模块的框架如图 24.13 所示。

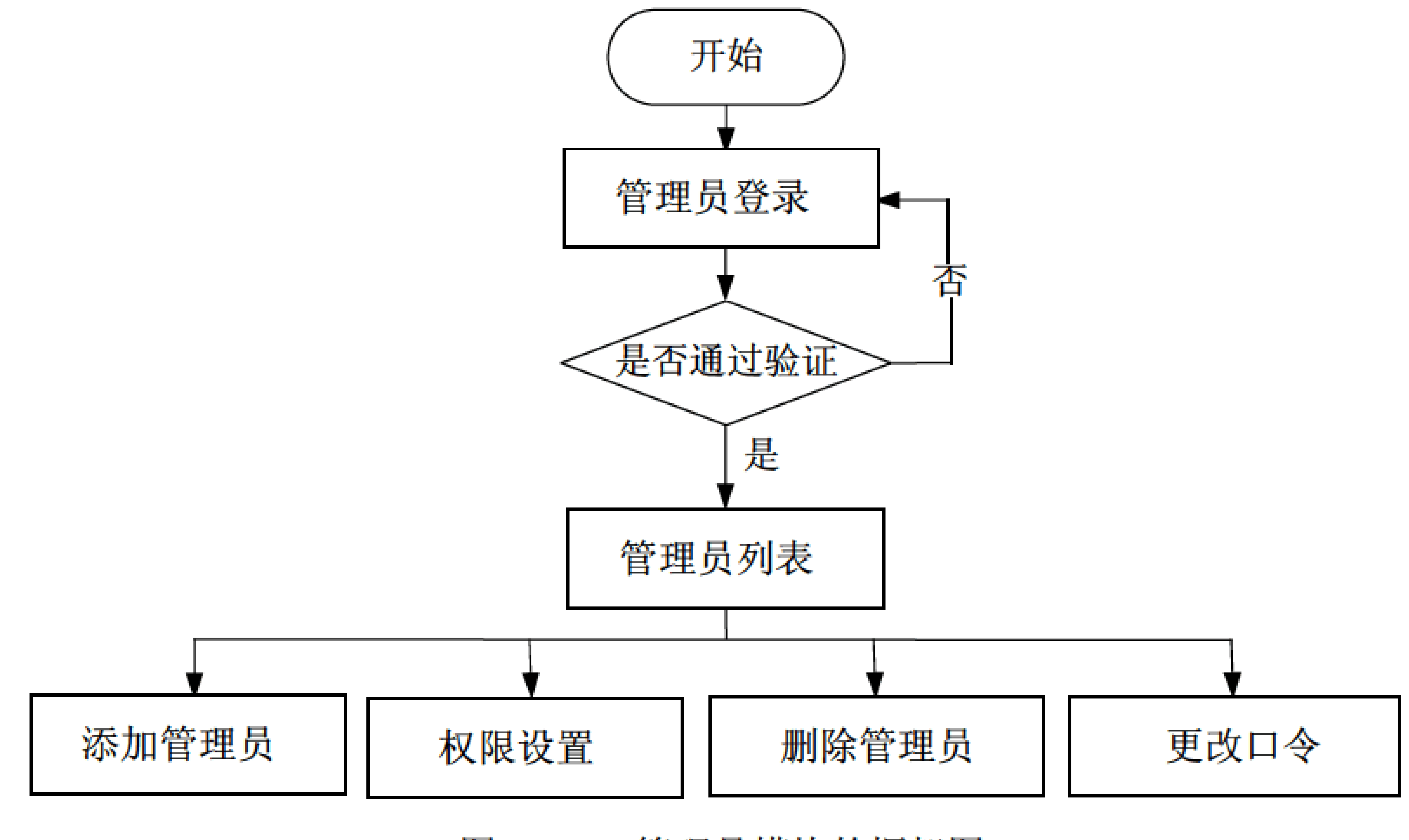


图 24.13　管理员模块的框架图

24.6.2 管理员模块技术分析

在管理员模块中，涉及的数据表是 tb_manager（管理员信息表）和 tb_purview（权限表）。其中，管理员信息表中保存的是管理员名称和密码等信息，权限表中保存的是各管理员的权限信息，这两个表通过各自的 id 字段相关联。通过这两个表可以获得完整的管理员信息。

注意：在实现系统登录前，需要在 MySQL 数据库中手动添加一条系统管理员的数据（管理员名为 mr，密码为 mrsoft，拥有所有权限），即在 MySQL 的客户端命令行中应用下面的语句分别向管理员信息表 tb_manager 和权限表 tb_purview 中各添加一条数据。

```
#添加管理员信息
insert into tb_manager (name,pwd) values('mr','mrsoft');
#添加权限信息
insert into tb_purview values(1,1,1,1,1,1);
```

从网站安全的角度考虑，仅有上面介绍的系统登录页面并不能有效地保存系统的安全，一旦系统首页面的地址被他人获得，就可以通过在地址栏中输入系统的首页面地址而直接进入到系统中。为了便于网站的维护，因此将验证用户是否登录的代码封装在独立的 PHP 文件中，即 check_login.php 文件。验证用户是否登录的具体代码如下：

代码位置：光盘\mr\24\check_login.php

```
<?php
session_start();                                          //初始化 session 变量
if($_SESSION[admin_name]==""){                            //如果 session 变量为空，则说明用户未登录
    echo "<script>alert('对不起，请通过正确的途径登录博考图书馆管理系统!');window.location.href='login.
php';</script>";
}
?>
```

当系统调用首页时，会判断 session 变量 admin_name 是否存在，如果不存在，将页面重定向到系统登录（login.php）页面。

24.6.3 系统登录的实现过程

系统登录是进入学校图书馆管理系统的入口，主要用于验证管理员的身份。运行本系统，首先进入的是系统登录页面，在该页面中，系统管理员可以通过输入正确的管理员名称和密码登录到系统首页，当用户没有输入管理员名称或密码时，系统会通过 JavaScript 进行判断，并给予信息提示。系统登录页面的运行结果如图 24.14 所示。



图 24.14 系统登录页面的运行结果



系统登录页面主要用于收集管理员的输入信息及通过自定义的 JavaScript 函数验证输入信息是否为空。该页面中所涉及的表单元素如表 24.1 所示。

表 24.1　系统登录页面所涉及的表单元素

名　称	元素类型	重要属性	含　义
form1	form	method="post" action="chklogin.php"	管理员登录表单
name	text	size="25"	管理员名称
pwd	password	size="25"	管理员密码
submit	submit	value="确定" onclick="return check(form1)"	“确定”按钮
submit3	reset	value="重置"	“重置”按钮
submit2	button	value="关闭" onClick="window.close();"	“关闭”按钮

编写自定义的 JavaScript 函数，用于判断管理员名称和密码是否为空。代码如下：

代码位置：光盘\mr\24\login.php

```
<script language="javascript">
function check(form){                        //自定义一个 JavaScript 函数 check()
    if (form.name.value==""){                //如果管理员名称为空，则弹出提示信息，并重新返回焦点
        alert("请输入管理员名称!");form.name.focus();return false;
    }
    if (form.pwd.value==""){                 //如果管理员密码为空，则弹出提示信息，并重新返回焦点
        alert("请输入密码!");form.pwd.focus();return false;
    }
}
</script>
```

提交表单到数据处理页，页面中为了防止非法用户进入学校图书馆管理系统首页，通过调用类的 chkinput()方法实现判断用户名和密码是否正确。如果为合法用户，则可以登录学校图书馆管理系统的首页；否则，弹出相应的错误提示。关键代码如下：

代码位置：光盘\mr\24\chklogin.php

```
<?php
session_start();                              //初始化 session 变量
$A_name=$_POST[name];                         //接收表单提交的用户名
$A_pwd=$_POST[pwd];                           //接收表单提交的密码
❶  class chkinput{                            //定义类
   var $name;
   var $pwd;
   function chkinput($x,$y){                  //定义一个方法
     $this->name=$x;                          //将管理员名称传给类对象$this->name
     $this->pwd=$y;                           //将管理员密码传给类对象$this->pwd
    }
   function checkinput(){
     include("conn/conn.php");                //连接数据库文件
     $sql=mysql_query("select * from tb_manager where name='".$this->name."' and pwd='".$this->pwd."'",
$conn);
     $info=mysql_fetch_array($sql);           //检索管理员名称和密码是否正确
     if($info==false){                        //如果管理员名称或密码不正确，则弹出相关提示信息
          echo "<script language='javascript'>alert('您输入的管理员名称错误，请重新输入！
');history.back();</script>";
```

```
                exit;
            }
        else{                                                    //如果管理员名称或密码正确，则弹出相关提示信息
            echo "<script>alert('管理员登录成功!');window.location='index.php';</script>";
                $_SESSION[admin_name]=$info[name];        //将管理员名称存到$_SESSION[admin_name]变量中
                $_SESSION[pwd]=$info[pwd];                //将管理员密码存到$_SESSION[pwd]变量中
        }
    }
}
❷          $obj=new chkinput(trim($name),trim($pwd));  //创建对象
❸      $obj->checkinput();                              //调用类
?>
```

代码导读：

❶ class：创建一个 PHP 类时，必须使用关键字 class 进行声明，该关键字后紧跟类的名称，之后用大括号将类体进行封装。

❷ new：PHP 中应用 new 关键字创建对象。该模块创建了一个名为 chkinput 的验证管理员类。

❸ $obj->checkinput();：通过$obj 对象调用 checkinput 类中的属性和方法，即管理员名称和密码。

24.6.4　查看管理员的实现过程

管理员登录后，选择“系统设置”/“管理员设置”菜单项，进入到查看管理员列表页面。在该页面中，将以表格的形式显示全部管理员及其权限信息，并提供添加管理员信息、删除管理员信息和设置管理员权限的超链接。查看管理员列表页面的运行结果如图 24.15 所示。

图 24.15　查看管理员列表页面的运行结果

首先使用左外联接语句（left join…on）从数据表 tb_manager 和 tb_purview 中查询出符合条件的数据，然后将查询结果应用 do…while 循环语句输出到浏览器。关键代码如下：

代码位置：光盘\mr\24\manager.php

```
<?php
include("conn/conn.php");                                          //连接数据库文件
❶ $sql=mysql_query("select m.id,m.name,p.sysset,p.readerset,p.bookset,p.borrowback,p.sysquery from tb_manager as m left join (select * from tb_purview) as p on m.id=p.id");
$info=mysql_fetch_array($sql);                                     //检索数据信息
```

```
do{                                                    //应用 do…while 循环语句输出查询结果
?>
/* ******************************输出符合查询条件的记录******************************** */
<tr>
    <td style="padding:5px;"><?php echo $info[name];?></td>
❷   <td align="center"><input name="checkbox" type="checkbox" class="noborder" value="checkbox"
disabled="disabled" <?php if($info[sysset]==1){echo ("checked");}?>></td>
    <td align="center"><input name="checkbox" type="checkbox" class="noborder" value="checkbox"
 disabled="disabled" <?php if($info[readerset]==1){echo("checked");}?>></td>
    <td align="center"><input name="checkbox" type="checkbox" class="noborder" value="checkbox" disabled
<?php if($info[bookset]==1){echo("checked");}?>></td>
    <td align="center"><input name="checkbox" type="checkbox" class="noborder" value="checkbox" disabled
<?php if($info[borrowback]==1){echo("checked");}?>></td>
    <td align="center"><input name="checkbox" type="checkbox" class="noborder" value="checkbox" disabled
<?php if($info[sysquery]==1){echo("checked");}?>></td>
    <td align="center"><a href="#" onClick="window.open('manager_modify.php?id=<?php echo
$info[id]; ?>','','width=292,height=175')">权限设置</a></td>
    <td align="center"><a href="manager_del.php?id=<?php echo $info[id];?>">删除</a></td>
  </tr>
/* ********************************************************************************** */
<?php
  }while($info=mysql_fetch_array($sql));                         //do…while 循环语句结束
?>
```

代码导读：

❶ left join…on：应用左外联接将两个表或多个表连接起来，返回部分或全部匹配行。

❷ <?php if($info[sysset]==1){echo ("checked");}?>：如果系统设置字段的值为 1，则复选框处于选中状态。

24.6.5　添加管理员的实现过程

管理员登录后，选择“系统设置”/“管理员设置”菜单项，进入到查看管理员列表页面，在该页面中单击“添加管理信息”超链接，打开添加管理员信息页面。添加管理员信息页面的运行结果如图 24.16 所示。

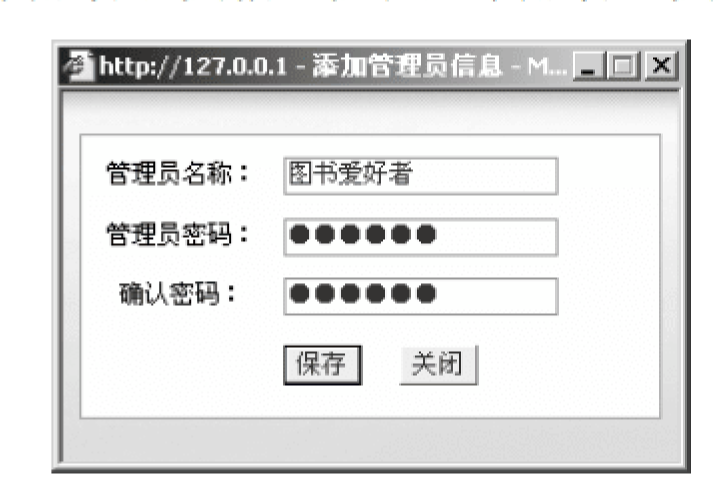


图 24.16　添加管理员页面的运行结果

注意：新添加的管理员信息没有权限，必须通过设置管理员权限为其指定可操作的功能模块。

在查看管理员列表页面中单击“添加管理信息”超链接文字的 HTML 代码如下：

代码位置：光盘\mr\24\manager.php

```
<a href="#" onClick="window.open('manager_add.php','','width=292,height=175')">添加管理员信息</a>
```

添加管理员页面主要用于收集输入的管理员信息及通过自定义的 JavaScript 函数验证输入信息是否合

法。该页面中所涉及的表单元素如表 24.2 所示。

表 24.2　添加管理员页面所涉及的表单元素

名　　称	元 素 类 型	重 要 属 性	含　　义
form1	form	method="post" action="manager_ok.php"	表单
name	text		管理员名称
pwd	password		管理员密码
pwd1	password		确认密码
submit	submit	value="保存" onClick="check(form1)"	“保存”按钮
Submit2	button	value="关闭" onClick="window.close();"	“关闭”按钮

在添加管理员页面中，输入合法的管理员名称及密码后，单击“保存”按钮，提交表单信息到数据处理页，将添加的管理员信息保存到数据表中。如果添加成功，弹出成功的提示信息；否则，弹出错误提示。代码如下：

代码位置：光盘\mr\24\manager_ok.php

```
<?php
include("conn/conn.php");                              //连接数据库文件
if($_POST[submit]!=""){                                //如果单击了“保存”按钮，则执行下面的操作
$name=$_POST[name];                                    //获取管理员名称
$pwd=$_POST[pwd];                                      //获取管理员密码
$sql=mysql_query("insert into tb_manager (name,pwd) values('$name','$pwd')");
if($sql==true){                                        //向数据表中添加管理员信息成功，则给出提示信息
echo "<script language=javascript>alert('管理员添加成功！');window.close();window.opener.location.reload(); </script>";
}
else{                                                  //向数据表中添加管理员信息失败，则给出提示信息
echo "<script language=javascript>alert('管理员添加失败！');window.close();window.opener.location.reload();</script>";
}
}
?>
```

技巧：在添加管理员处理页中应用 window.opener.location.reload();语句刷新父窗口中的信息。

24.6.6　设置管理员权限的实现过程

在查看管理员列表页面中单击指定管理员后面的“权限设置”超链接，即可进入到“权限设置”页面，设置该管理员的操作权限。权限设置页面的运行结果如图 24.17 所示。

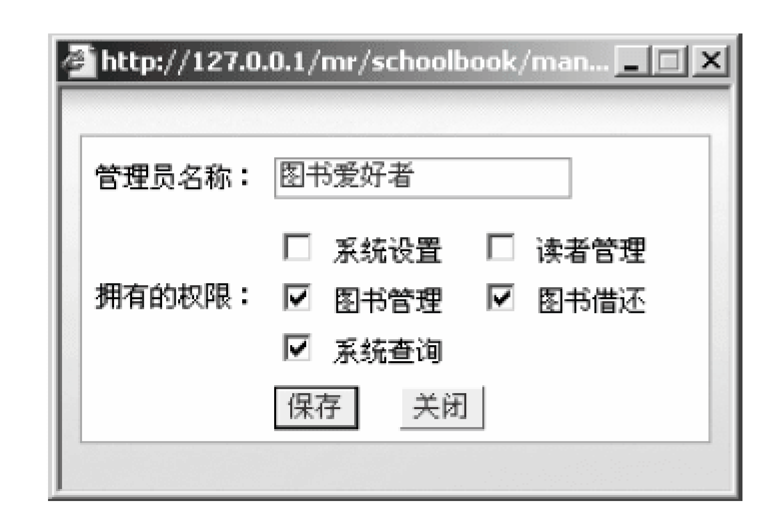
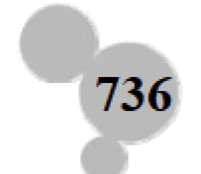


图 24.17　权限设置页面的运行结果



权限设置页面中所涉及的表单元素如表 24.3 所示。

表 24.3　权限设置页面所涉及的表单元素

名　称	元 素 类 型	重 要 属 性	含　义
form1	form	method="post" action="manager_modifyok.php"	表单
id	hidden	value="<?php echo $info[id];?>"	管理员编号
name	text	value="<?php echo $info[name];?>"	管理员名称
sysset	checkbox	<?php if($info[sysset]==1){ echo("checked");}?>	系统设置
readerset	checkbox	<?php if($info[readerset]==1){ echo("checked");}?>	读者管理
bookset	checkbox	<?php if($info[bookset]==1){echo("checked");}?>	图书管理
borrowback	checkbox	<?php if($info[borrowback]==1){echo("checked");}?>	图书借还
sysquery	checkbox	<?php if($info[sysquery]==1){echo("checked");}?>	系统查询
submit	submit	class="btn_grey"	“保存”按钮
submit2	button	value="关闭" onClick="window.close();"	“关闭”按钮

在查看管理员列表页面中添加“权限设置”列，并在该列中添加以下用于打开“权限设置”页面的超链接代码。

代码位置：光盘\mr\24\manager.php

```
<a href="#" onClick="window.open('manager_modify.php?id=<?php echo $info[id]; ?>','','width=292,height= 175')">
权限设置</a>
```

从上面的 URL 地址中可以获取设置管理员权限页所涉及的 id 号，将 id 号提交给处理页 manager_modifyok.php，修改 id 号所对应的管理员信息。具体代码如下：

代码位置：光盘\mr\24\manager_modifyok.php

```
<?php
include("conn/conn.php");                                   //连接数据库文件
if($_POST[submit]!=""){                                     //如果提交表单，则执行以下操作
$id=$_POST[id];                                             //获取 id 信息
❶    $sysset=$_POST[sysset]==""?0:1;                        //应用三目运算符求出“系统设置”复选框的值
$readerset=$_POST[readerset]==""?0:1;                       //应用三目运算符求出“读者管理”复选框的值
$bookset=$_POST[bookset]==""?0:1;                           //应用三目运算符求出“图书管理”复选框的值
$borrowback=$_POST[borrowback]==""?0:1;                     //应用三目运算符求出“图书借还”复选框的值
$sysquery=$_POST[sysquery]==""?0:1;                         //应用三目运算符求出“系统查询”复选框的值
$query=mysql_query("select * from tb_purview where id=$id");
        $info=mysql_fetch_array($query);                    //检索权限信息表中是否存在该管理员
if($info==false){                                           //如果不存在，向权限表中添加管理员权限信息
❷    mysql_query("insert into tb_purview(id,sysset,readerset,bookset,borrowback,sysquery)
values($id,$sysset,$readerset,$bookset,$borrowback,$sysquery)");
}
else{                                                       //否则，更新管理员的权限信息
        mysql_query("update tb_purview set
 sysset=$sysset,readerset=$readerset,bookset=$bookset,borrowback=$borrowback,sysquery=$sysquery
where id='$id'");
}
echo"<script  language=javascript>alert('权限设置修改成功！ ');window.close();window.opener.location.reload();
</script>";                                                 //更新成功，弹出提示信息，并更新父窗口
}
?>
```

代码导读：

❶ $_POST[sysset]==""?0:1;：应用三目运算符求出“系统设置”复选框的值，如果等于空，值为 0，否则值为 1。

❷ insert…into：向权限信息表中添加一行数据信息，insert…into 语句只适用于对单行数据的插入。

24.6.7　删除管理员的实现过程

删除管理员使用的数据表：tb_manager、tb_purview

在查看管理员列表页面中单击指定管理员信息后面的“删除”超链接，该管理员及其权限信息将被删除。

在查看管理员列表页面中添加以下用于删除管理员信息的超链接代码。

代码位置：光盘\mr\24\manager.php

```
<a href="manager_del.php?id=<?php echo $info[id];?>">删除</a>
```

从上面的 URL 地址中，可以获取删除管理员所涉及的 id 号，将 id 号提交给 manager_del.php 处理页删除 id 号所对应的管理员信息。具体代码如下：

代码位置：光盘\mr\24\manager_del.php

```
<?php
include("conn/conn.php");                                          //连接数据库文件
$id=$_GET[id];                                                     //获取管理员的 id 号
$sql=mysql_query("delete from tb_manager where id='$id'");         //删除管理员表中 id 号所对应的管理员信息
$query=mysql_query("delete from tb_purview where id='$id'");       //删除权限表中 id 号所对应的管理员权限
if($sql==true and $query==true ){                                  //如果删除操作成功，则弹出提示信息
    echo "<script language=javascript>alert('管理员删除成功！');history.back();</script>";
}
else{                                                              //如果删除操作失败，则弹出提示信息
    echo "<script language=javascript>alert('管理员删除失败！');history.back();</script>";
}
?>
```

24.7　图书档案管理模块设计

24.7.1　图书档案管理模块概述

图书档案管理模块主要包括查看图书列表、添加图书信息、修改图书信息、删除图书信息和查看图书详细信息 5 个功能。图书档案模块的框架如图 24.18 所示。

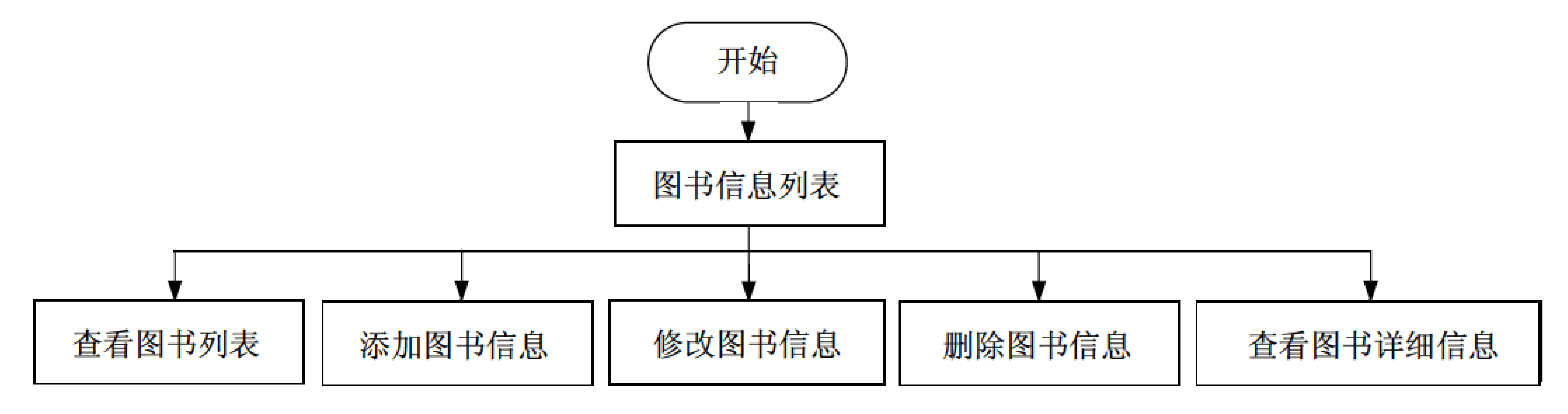


图 24.18　图书档案模块的框架图



24.7.2　图书档案管理模块技术分析

在图书档案管理模块中，涉及的数据表是 tb_bookinfo（图书信息表）、tb_bookcase（书架设置表）、tb_booktype（图书类型表）和 tb_publishing（出版社信息表），这 4 个数据表间通过相应的字段进行关联，如图 24.19 所示，通过以上 4 个表可以获得完整的图书档案信息。

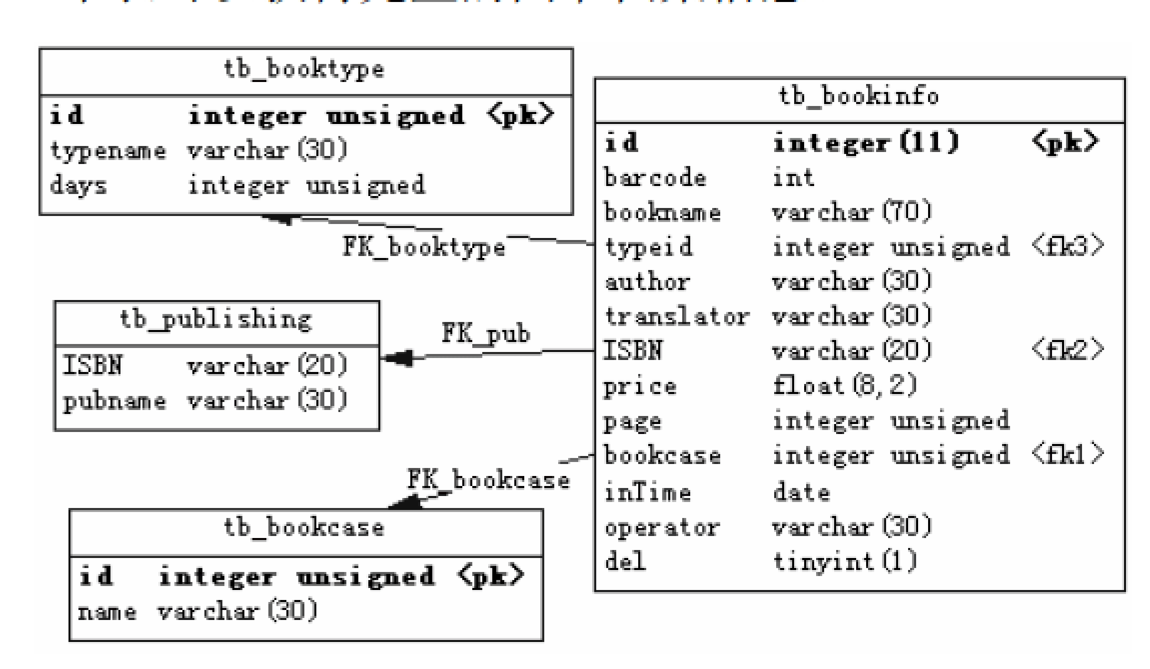


图 24.19　图书档案管理模块各表间的关系图

注意：本模块主要应用联接语句实现多表查询。

24.7.3　查看图书信息列表的实现过程

管理员登录后，选择“图书管理”/“图书档案管理”菜单项，进入到查看图书列表页面，在该页面中将显示全部图书信息列表，同时提供添加图书信息、删除图书信息、修改图书信息的超链接。查看图书信息列表页面的运行结果如图 24.20 所示。



图 24.20　查看图书信息列表页面的运行结果

打开功能导航 navigation.php 文件，设置“图书档案管理”菜单项的超链接的代码如下：

代码位置：光盘\mr\24\navigation.php

```
<a href="book.php" class="a1">图书档案管理</a>
```

首先应用 join…on 内联接语句将 tb_bookinfo、tb_bookcase、tb_booktype 和 tb_publishing 4 个数据表连接起来检索指定条件的图书信息，然后应用 do…while 循环语句输出查询结果到浏览器。查看图书信息页面的代码如下：

代码位置：光盘\mr\24\book.php

```
<?php
include("conn/conn.php");                                    //连接数据库文件
$query=mysql_query("select book.barcode,book.id as bookid,book.bookname,bt.typename,pb.pubname,bc.name
from tb_bookinfo book join tb_booktype bt on book.typeid=bt.id join tb_publishing pb on book.ISBN=pb.ISBN
join tb_bookcase bc on book.bookcase=bc.id");
$result=mysql_fetch_array($query);                           //应用外联接检索图书信息
?>
…                                                            //省略图书信息标题 HTML 标记部分
<?php
 do{                                                         //应用 do…while 循环语句输出查询结果
?>
  <tr>
    <td style="padding:5px;"> <?php echo $result[barcode];?></td>
    <td style="padding:5px;"><a href="book_look.php?id=<?php echo $result[bookid];?>"><?php echo
$result[bookname];?></a></td>
    <td style="padding:5px;"> <?php echo $result[typename];?></td>
    <td style="padding:5px;"> <?php echo $result[pubname];?></td>
    <td style="padding:5px;"> <?php echo $result[name];?></td>
    <td align="center"><a href="book_Modify.php?id=<?php echo $result[bookid];?>">修改</a></td>
    <td align="center"><a href="book_del.php?id=<?php echo $result[bookid];?>">删除</a></td>
  </tr>
<?
  }while($result=mysql_fetch_array($query));                 //do…while 循环语句结束
?>
```

24.7.4　添加图书信息的实现过程

管理员登录系统后，在导航栏中单击“图书档案管理”超链接，进入到查看图书列表页面。在该页面中单击“添加图书信息”超链接，进入到添加图书信息页面。添加图书信息页面的运行结果如图 24.21 所示。

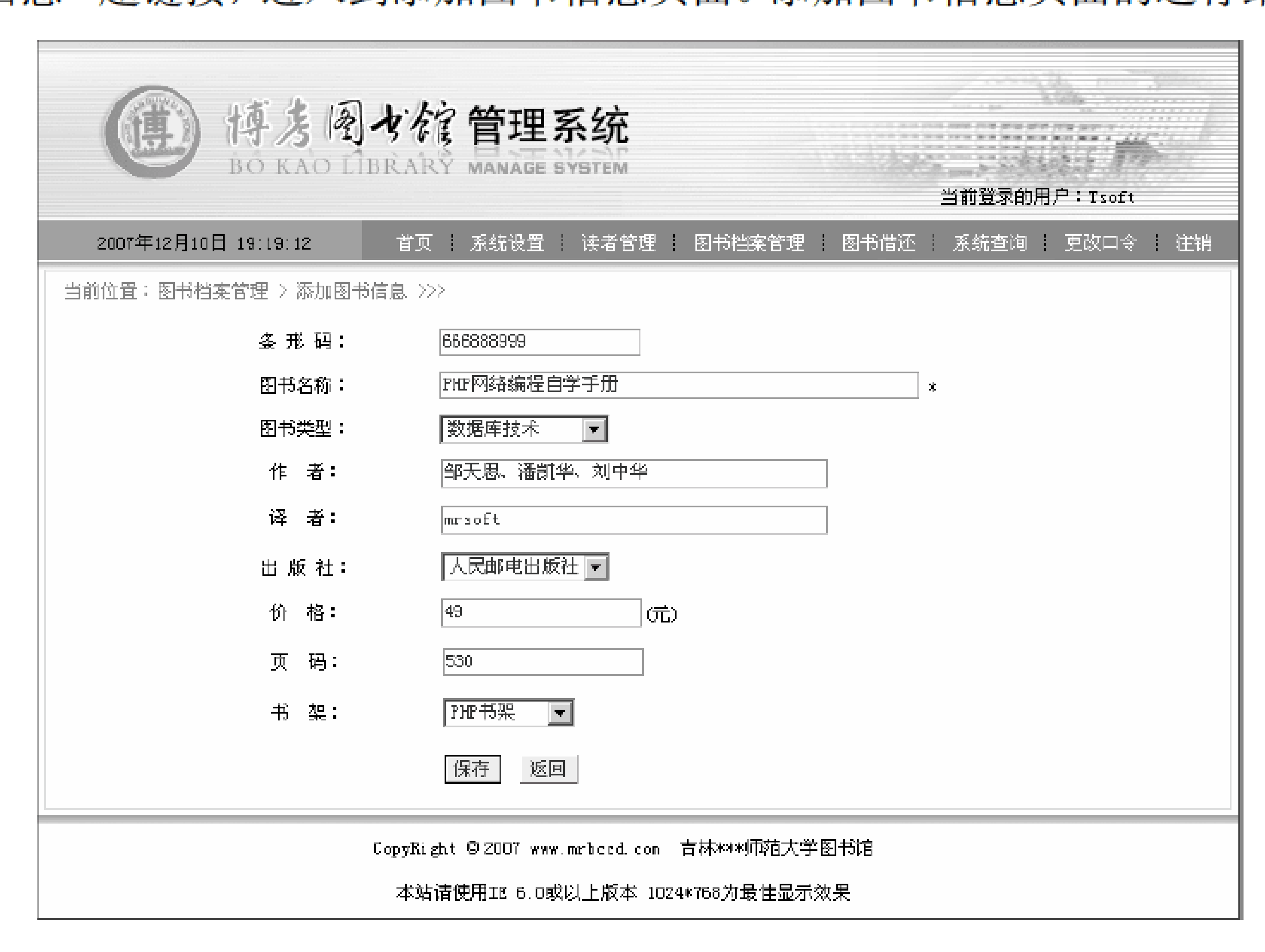


图 24.21　添加图书信息页面的运行结果

在查看图书列表页面中设置“添加图书信息”超链接的代码如下：

代码位置：光盘\mr\24\book.php

```
<a href="book_add.php">添加图书信息</a>
```

添加图书信息页面主要用于收集输入的图书信息以及通过自定义的 JavaScript 函数验证输入信息是否合法。该页面中所涉及的重要表单元素如表 24.4 所示。

表 24.4　添加图书信息页面所涉及的重要表单元素

名　　称	元 素 类 型	重 要 属 性	含　　义
form1	form	method="post" action="book_ok.php"	表单
typeId	select	<?php include("Conn/conn.php"); $sql=mysql_query("select * from tb_booktype"); $info=mysql_fetch_array($sql); do{ ?> <option value="<?php echo $info[id];?>"> <?php echo $info[typename];?></option> <?php }while($info=mysql_fetch_array($sql));?>	图书类型
isbn	select	<?php $sql2=mysql_query("select * from tb_publishing"); $info2=mysql_fetch_array($sql2); do{ ?> <option value="<?php echo $info2[ISBN];?>"> <?php echo $info2[pubname];?></option> <?php }while($info2=mysql_fetch_array($sql2));?>	出版社
bookcaseid	select	<?php $sql3=mysql_query("select * from tb_bookcase"); $info3=mysql_fetch_array($sql3); do{ ?> <option value="<?php echo $info3[id];?>"> <?php echo $info3[name];?></option> <?php }while($info3=mysql_fetch_array($sql3));?>	书架名称
operator	hidden	value="<?php echo $info3[name];?>"	操作员
submit	submit	onClick="return check(form1)"	“保存”按钮
submit2	button	onClick="history.back();"	“返回”按钮

由于添加图书信息的方法同添加管理员信息的方法类似，所以此处只给出向图书信息表中插入数据的 SQL 语句，详细代码参见光盘。向图书信息表中插入数据的 SQL 语句如下：

代码位置：光盘\mr\24\book_ok.php

```
mysql_query("insert into
tb_bookinfo(barcode,bookName,typeid,author,translator,ISBN,price,page,bookcase,inTime,operator )values('$b
arcode','$bookName','$typeid','$author','$translator','$isbn','$price','$page','$bookcaseid','$inTime','$operator')");
```

24.7.5　修改图书信息的实现过程

管理员登录系统后，在导航栏中单击“图书档案管理”超链接，进入到查看图书列表页面。单击想要修改的图书信息后面的“修改”超链接，进入到“修改图书信息”页面。修改图书信息页面的运行结果如图 24.22 所示。



图 24.22　修改图书信息页面的运行结果

在图书信息列表页面中，添加“修改”超链接的代码如下：

代码位置：光盘\mr\24\book.php

```
<a href="book_Modify.php?id=<?php echo $result[bookid];?>">修改</a>
```

在修改图书信息页面中修改图书信息后，单击“保存”按钮，提交表单信息到数据处理页 book_Modify_ok.php，应用 UPDATE 语句将修改的图书信息保存到数据表 tb_bookinfo 中，并弹出“图书信息修改成功！”提示信息，将页面重定向到修改图书信息页。数据处理页的代码如下：

代码位置：光盘\mr\24\book_Modify_ok.php

```
<?php
session_start();                                    //初始化 session 变量
include("conn/conn.php");                           //连接数据库文件
$bid=$_POST[bid];                                   //获取图书 id 号
$operator=$_SESSION[admin_name];                    //获取管理员名称
$barcode=$_POST[barcode];                           //获取图书条形码
$bookName=$_POST[bookName];                         //获取图书名称
$typeid=$_POST[typeId];                             //获取图书类型 id 号
$author=$_POST[author];                             //获取图书作者
$translator=$_POST[translator];                     //获取图书译者
$isbn=$_POST[isbn];                                 //获取出版社 ISBN
$price=$_POST[price];                               //获取图书单价
$page=$_POST[page];                                 //获取图书页码
$bookcase=$_POST[bookcaseid];                       //获取图书书架 id 号
```

```
$inTime=date("Y-m-d");                                    //设置图书更新日期为当前日期
$query=mysql_query("update tb_bookinfo set barcode='$barcode', bookName='$bookName' , typeid='$typeid',
author='$author', translator='$translator', ISBN='$isbn' , price='$price' , page='$page' , bookcase='$bookcaseid',
inTime='$inTime', operator='$operator' where id=$bid");   //更新数据表
echo "<script language='javascript'>alert('图书信息修改成功!');history.back();</script>";
?>
```

24.7.6　删除图书信息的实现过程

在查看图书列表页面中，设置“删除”超链接的代码如下：

代码位置：光盘\mr\24\book.php

```
<a href="book_del.php?id=<?php echo $result[bookid];?>">删除</a>
```

单击想要删除的图书信息后面的“删除”超链接，提交表单信息到数据处理页 book_del.php，应用 DELETE 语句将指定的图书信息从数据表 tb_bookinfo 中删除，如果删除操作执行成功，则弹出“图书信息删除成功！”提示信息，并将页面重定向到图书信息列表页面。数据处理页的代码如下：

代码位置：光盘\mr\24\book_del.php

```
<?php
include("conn/conn.php");                                          //连接数据库文件
$info_del=mysql_query("delete from tb_bookinfo where id=$_GET[id]");  //删除指定的图书信息
if($info_del){                                                     //如果信息删除成功，则弹出提示
    echo "<script language='javascript'>alert('图书信息删除成功!');history.back();</script> ";
}
?>
```

24.8　图书借还模块设计

24.8.1　图书借还模块概述

图书借还模块主要包括图书借阅、图书续借、图书归还、图书档案查询、图书借阅查询和借阅到期提醒 6 个功能。在图书借阅模块中的用户只有一种身份，那就是操作员，通过该身份可以进行图书借还等相关操作。图书借还模块的用例图如图 24.23 所示。

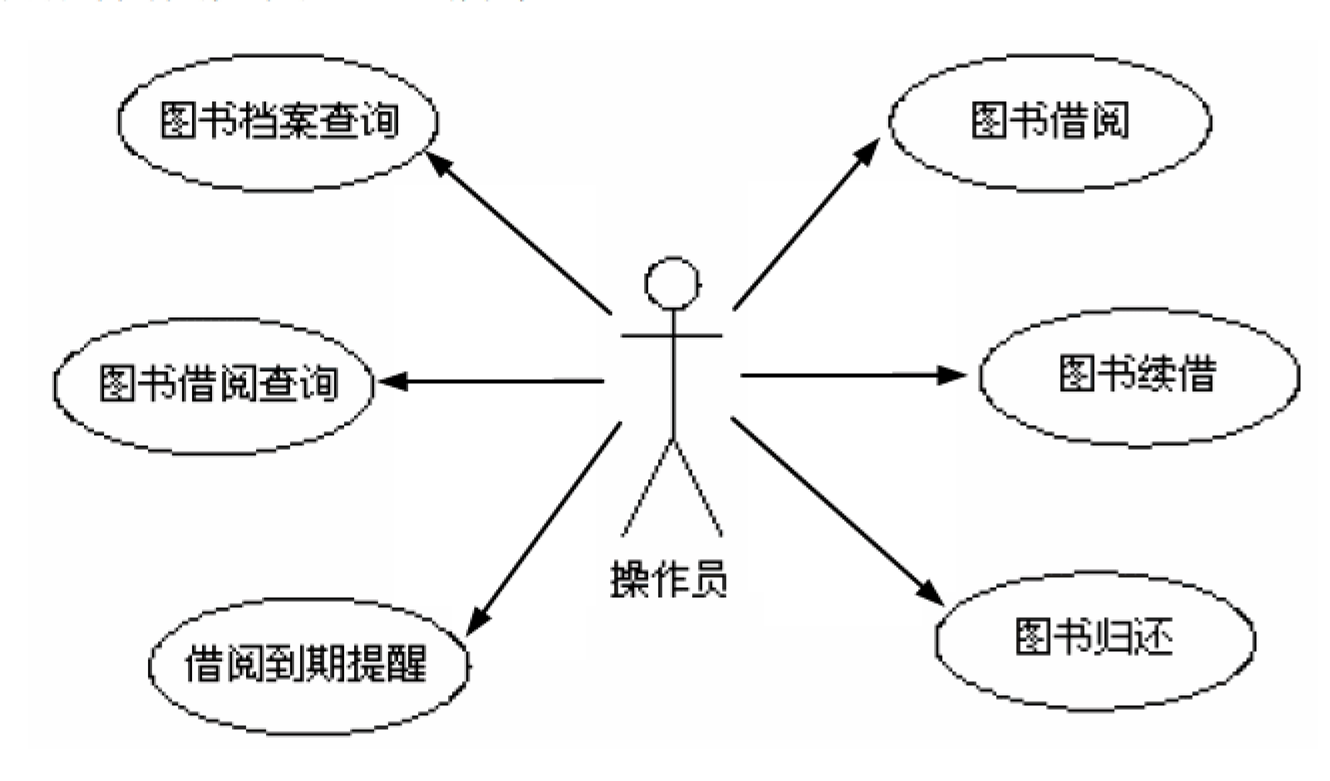


图 24.23　图书借还模块的用例图

24.8.2　图书借还模块技术分析

在图书借还模块中涉及的数据表是 tb_borrow（图书借阅信息表）、tb_bookinfo（图书信息表）和 tb_reader（读者信息表），这 3 个数据表间通过相应的字段进行关联，如图 24.24 所示。

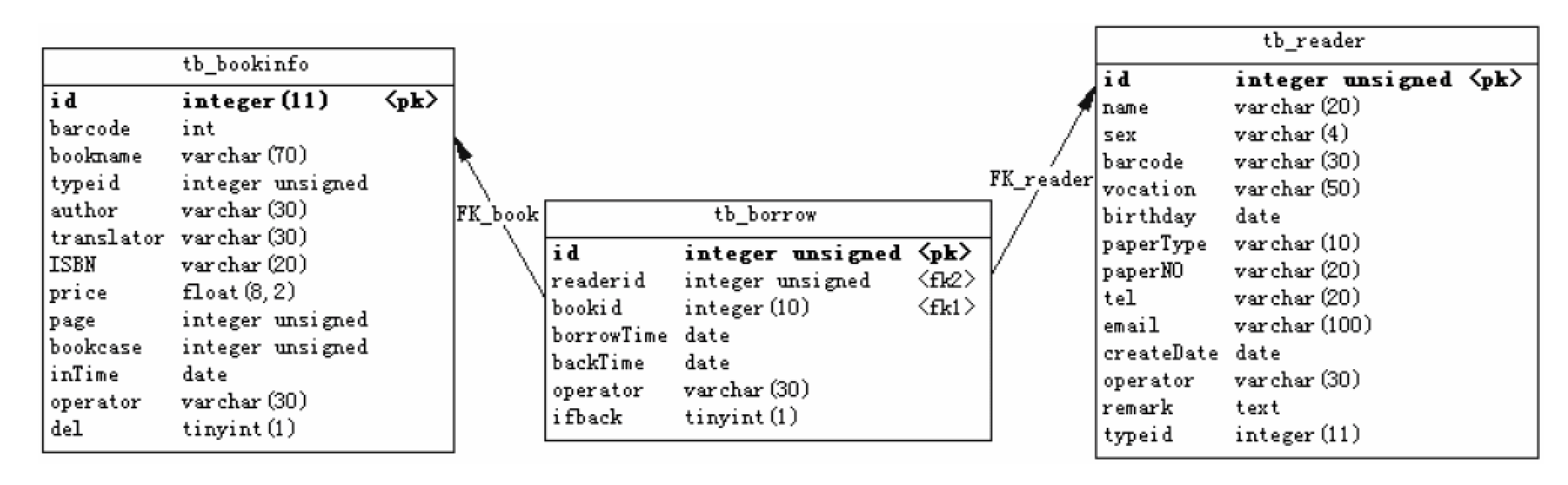


图 24.24　图书借还管理模块各表间关系图

24.8.3　图书借阅的实现过程

图书借阅使用的数据表：tb_borrow、tb_bookinfo、tb_reader

管理员登录后，选择“图书借还”/“图书借阅”菜单项，进入到图书借阅页面，在该页面的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的借阅图书信息。如果检索到对应的读者信息，则将其显示在页面中，此时输入图书的条形码或图书名称后，单击“确定”按钮，借阅指定的图书，运行结果如图 24.25 所示。



图 24.25　图书借阅页面的运行结果

说明：当读者借阅图书完毕后，操作员通过单击“完成借阅”按钮，将重新载入图书借阅页面，当前页处于空信息状态，从而方便操作员进行下一个读者借阅图书操作。



图书借阅页面总体上可以分为两个部分：一部分用于查询并显示读者信息，另一部分用于显示读者的借阅信息和添加读者借阅信息。图书借阅页面在 Dreamweaver 中的设计效果如图 24.26 所示。



图 24.26　图书借阅页面的设计效果

在进行图书借阅时，系统要求每个读者只能同时借阅一定数量的图书，并且该数量由读者类型表 tb_readerType 中的可借数量 number 决定，所以笔者编写了自定义的 checkbook()函数，用于判断当前选择的读者是否还可以借阅新的图书，同时该函数还具有判断输入读者条形码或图书名称文本框是否为空的功能。代码如下：

代码位置：光盘\mr\24\bookBorrow.php

```
<script language="javascript">
function checkbook(form){                                    //自定义一个 JavaScript 函数 checkbook()
    if(form.barcode.value==""){                              //如果读者条形码为空
        alert("请输入读者条形码!");form.barcode.focus();return;    //弹出提示，焦点返回到条形码文本框
    }
    if(form.inputkey.value==""){                             //如果图书查询文本框的值为空
        alert("请输入查询关键字!");form.inputkey.focus();return;   //弹出提示，焦点返回到图书查询文本框
    }
    if(form.number.value-form.borrowNumber.value<=0){        //如果图书的借阅数量超过了可借数量
            alert("您不能再借阅其他图书了!");return;          //弹出提示信息
    }
  form.submit();                                             //提交表单
 }
</script>
```

技巧： 在 JavaScript 中比较两个数值型文本框的值时，不使用运算符“==”，而是将这两个值相减，再判断其结果。

检索读者的基本信息和未归还的借阅图书信息的 SQL 语句如下：

代码位置：光盘\mr\24\bookBorrow.php

```
$sql=mysql_query("select r.*,t.name as typename,t.number from tb_reader r left join tb_readerType t on
r.typeid=t.id where r.barcode='$barcode'");
$info=mysql_fetch_array($sql);                               //检索读者信息
```

获取读者借阅信息的 SQL 语句如下：

代码位置：光盘\mr\24\bookBorrow.php

```
$sql1=mysql_query("select
r.*,borr.borrowTime,borr.backTime,book.bookname,book.price,pub.pubname,bc.name as
bookcase from tb_borrow as borr join tb_bookinfo as book on book.id=borr.bookid join tb_publishing as pub on
book.ISBN=pub.ISBN   join tb_bookcase as bc on book.bookcase=bc.id join tb_reader as r on borr.readerid=r.id
where
```

```
 borr.readerid='$readerid' and borr.ifback=0");
$info1=mysql_fetch_array($sql1);                       //检索读者的借阅信息
$borrowNumber=mysql_num_rows($sql1);                   //获取结果集中行的数目
```

在“图书条形码”/“图书名称”文本框中输入图书条形码或图书名称后，单击“确定”按钮，检索图书信息是否存在，如果不存在，则向图书借阅信息表中添加该读者的图书的借阅记录，完成图书借阅操作；否则，弹出该书不能被同一读者重复借阅的提示信息。图书借阅的具体代码如下：

代码位置：光盘\mr\24\bookBorrow.php

```
 <?php
if($_POST[inputkey]!=""){                              //如果“图书条形码”/“图书名称”后的文本框不为空
$f=$_POST[f];                                          //获取用户选择的条件值
❶    $inputkey=trim($_POST[inputkey]);                //获取用户输入的查询关键字
$barcode=$_POST[barcode];                              //获取读者的条形码
$readerid=$_POST[readerid];                            //获取读者 id 号
❷    $borrowTime=date('Y-m-d');                       //图书的借阅时间为系统当前时间
❸    $backTime=date("Y-m-d",(time()+3600*24*30));     //归还图书日期为当前期日期+30 天期限
$query=mysql_query("select * from tb_bookinfo where $f='$inputkey'");
$result=mysql_fetch_array($query);                     //检索图书信息是否存在
if($result==false){                                    //如果读者借阅的图书不存在，那么弹出提示信息
     echo "<script language='javascript'>alert('该图书不存在！
');window.location.href='bookBorrow.php?barcode=$barcode'; </script>";
}
else{                                                  //检索该读者所借阅的图书是否与再借图书重复
     $query1=mysql_query("select
r.*,borr.borrowTime,borr.backTime,book.bookname,book.price,pub.pubname,bc.name as bookcase from
tb_borrow as borr join tb_reader as r on borr.readerid=r.id join tb_bookinfo as book on book.id=borr.bookid join
tb_publishing as pub on book.ISBN=pub.ISBN  join tb_bookcase as bc on book.bookcase=bc.id where
borr.bookid=$result[id] and borr.readerid=$readerid and ifback=0");
     $result1=mysql_fetch_array($query1);
     if($result1==true){                               //如果所借图书已被该读者借阅，那么提示不能重复借阅
          echo "<script language='javascript'>alert('该图书已经借阅！
');window.location.href='bookBorrow.php?barcode=$barcode';</script>";
     }
     else{                                             //否则，完成图书借阅操作，并弹出借阅成功提示信息
          $bookid=$result[id];                         //将读者 id 号赋给一变量
          mysql_query("insert into
tb_borrow(readerid,bookid,borrowTime,backTime,operator,ifback)values('$readerid','$bookid','$borrowTime',
'$backTime','$_SESSION[admin_name]',0)");                  //向借阅信息表中添加一条借阅信息
          echo "<script language='javascript'>alert('图书借阅操作成功！');
window.location.href='bookBorrow.php?barcode=$barcode';</script>";
     }
     }
}
?>
```

代码导读：

❶ trim()：删除字符串中首尾的空白或者其他字符，以达到精确查询。

❷ date('Y-m-d')：获取系统的当前日期为图书借阅的日期，并格式化日期格式。

❸ date("Y-m-d",(time()+3600*24*30))：归还图书日期=当前期日期+30 天期限。当前日期时间戳应用 time()函数获取，30 天期限的时间戳等于 3600 秒×24 小时×30 天，并通过 date()函数格式化为指定日期格式。

24.8.4　图书续借的实现过程

管理员登录后，选择“图书借还”/“图书续借”菜单项，进入到图书续借页面。在该页面的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的借阅图书信息。如果检索到对应的读者信息，则将其显示在页面中，此时单击“续借”超链接，即可续借指定图书（即将该图书的归还时间加上该书的可借天数 30 天计算得出）。图书续借页面的运行结果如图 24.27 所示。



图 24.27　图书续借页面的运行结果

> **说明：** 当读者续借完图书后，操作员通过单击“完成续借”按钮，将重新载入图书续借页面，当前页处于空信息状态，从而方便操作员进行下一个读者续借图书操作。

图书续借页面的设计方法同图书借阅类似，所不同的是，在图书续借页面中没有添加借阅图书的功能，而是添加了“续借”超链接。图书续借页面在 Dreamweaver 中的设计效果如图 24.28 所示。

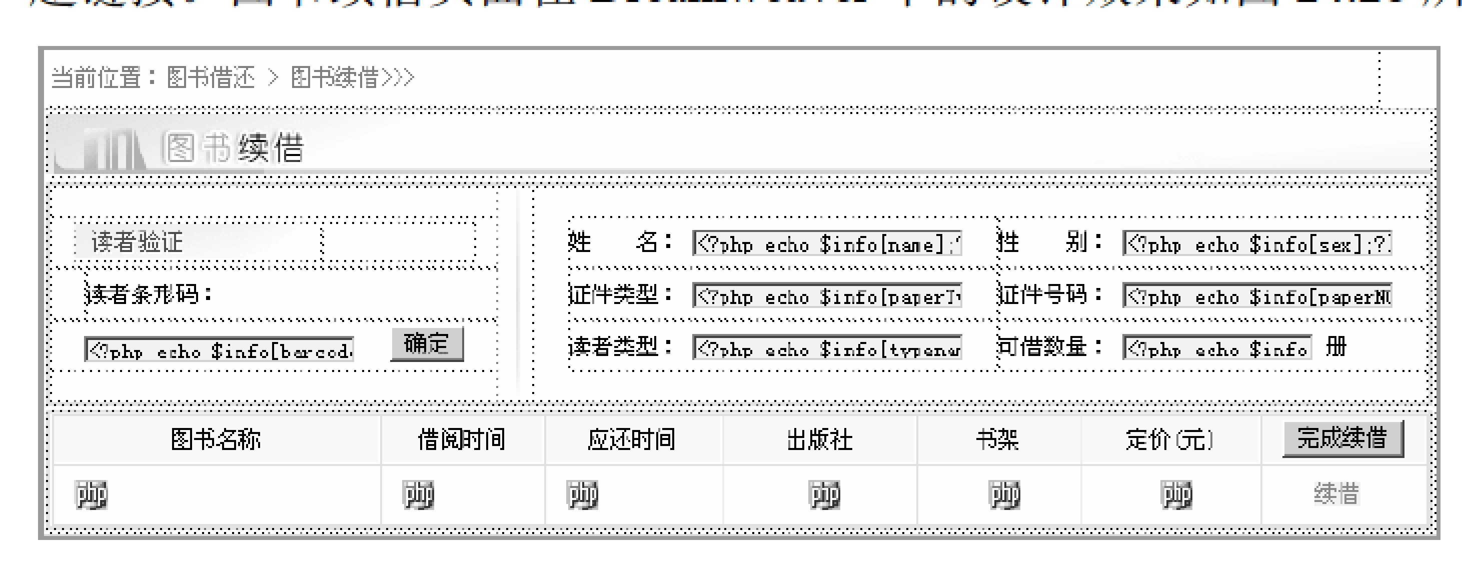


图 24.28　图书续借页面的设计效果

单击“续借”超链接时，还需要将读者条形码、借阅 id 号和图书归还时间一同传递到图书续借的处理页 borrow_oncemore.php 中。代码如下：

代码位置：光盘\mr\24\bookRenew.php

```
<a href="borrow_oncemore.php?barcode=<?php echo $info[barcode];?>&borrid=<?php echo
$info[borrid];?>&backTime=<?php echo $info[backTime];?>">续借</a>
```

检索读者信息和读者借阅信息的 SQL 语句如下：

代码位置：光盘\mr\24\bookRenew.php

```
$sql=mysql_query("select borr.id as borrid,borr.borrowTime,borr.backTime,borr.ifback,r.*,t.name as
typename,t.number,book.bookname,book.price,pub.pubname,bc.name as bookcase from tb_borrow as borr
join tb_reader r on
borr.readerid=r.id join tb_readerType t on r.typeid=t.id join tb_bookinfo as book on book.id=borr.bookid join
tb_publishing as pub on book.ISBN=pub.ISBN join tb_bookcase as bc on book.bookcase=bc.id where r.barcode=
'$barcode' and borr.ifback=0");
$info=mysql_fetch_array($sql);                                    //检索读者信息和借阅信息
```

单击“续借”超链接，提交到数据处理页 borrow_oncemore.php，主要用于完成图书的续借功能，它通过更改图书的归还日期（即将该图书的归还时间加上该书的可借天数 30 天计算得出）实现。数据处理页的代码如下：

代码位置：光盘\mr\24\borrow_oncemore.php

```
<?php
session_start();                                                  //初始化 session 变量
include("conn/conn.php");                                         //连接数据库文件
$barcode=$_GET[barcode];                                          //获取图书条形码
$new=$_GET[backTime];                                             //获取图书归还时间
//更新续借期，将动态获取的还书期日转化为时间截，然后再求出续借后的还书日期
$newbackTime=date("Y-m-d",(mktime(0, 0, 0, substr($new,5,2), substr($new,8,2), substr($new,0,4))+3600*24*30));
$borrid=$_GET[borrid];                                            //获取续借图书的 id 号
mysql_query("update tb_borrow set backTime='$newbackTime',ifback=0,operator='$_SESSION[admin_name]'
where
id=$borrid");
echo "<script language='javascript'>alert('图书续借操作成功！
');window.location.href='bookRenew.php?barcode=$barcode';</script>";    //弹出图书续借成功的提示信息
?>
```

24.8.5　图书归还的实现过程

管理员登录后，选择“图书借还”/“图书归还”菜单项，进入到图书归还页面。在该页面的“读者条形码”文本框中输入读者的条形码（如 123456789）后，单击“确定”按钮，系统会自动检索出该读者的基本信息和未归还的借阅图书信息。如果检索到对应的读者信息，则将其输出到浏览器，此时单击“归还”超链接，即可将指定图书归还。图书归还页面的运行结果如图 24.29 所示。



图 24.29　图书归还页面的运行结果

图书归还页面的设计方法同图书续借类似，所不同的是，将图书续借页面中的“续借”超链接转化为“归还”超链接。在单击“归还”超链接时，也需要将读者条形码和借阅 id 号一同传递到图书归还处理页。代码如下：

代码位置：光盘\mr\24\bookBack.php

```
<a href="bookBack_ok.php?borrid=<?php echo $info[borrid];?>&barcode=<?php echo $info[barcode];?>">归还
</a>
```

检索读者信息及读者借阅信息的 SQL 语句如下：

代码位置：光盘\mr\24\bookBack.php

```
$sql=mysql_query("select borr.id as borrid,borr.borrowTime,borr.backTime,borr.ifback,r.*,t.name as
typename,t.number,book.bookname,book.price,pub.pubname,bc.name as bookcase from tb_borrow as borr
join tb_reader r on borr.readerid=r.id join tb_readerType t on r.typeid=t.id join tb_bookinfo as book on book.id=
borr.bookid join tb_publishing as pub on book.ISBN=pub.ISBN  join tb_bookcase as bc on book.bookcase=
bc.id where r.barcode='$barcode' and borr.ifback=0");
$info=mysql_fetch_array($sql);                                    //检索读者信息及该读者的借阅信息
```

单击“归还”超链接，即可将指定图书归还。数据处理页的代码如下：

代码位置：光盘\mr\24\bookBack_ok.php

```
<?php
session_start();                                                  //初始化 session 变量
include("conn/conn.php");                                         //连接数据库文件
$backTime=date("Y-m-d");                                          //归还图书日期
$borrid=$_GET[borrid];                                            //获取读者的 id 号
mysql_query("update tb_borrow set backTime='$backTime',ifback=1,operator='$_SESSION[admin_name]'
where id=$borrid");                                               //更新读者的借阅信息
echo "<script language='javascript'>alert('图书归还操作成功！');window.location.href='bookBack.php?barcode=
$barcode';</script>";                                             ///弹出图书归还成功的提示信息
?>
```

24.8.6　图书借阅查询的实现过程

管理员登录后，选择“系统查询”/“图书借阅查询”菜单项，进入到图书借阅查询页面。图书借阅查询页面的运行结果如图 24.30 所示。在该页面中可以按指定的字段或某一时间段进行查询，同时还可以实现按指定字段及时间段进行综合条件查询。



图 24.30　图书借阅查询页面的运行结果

图书借阅查询页面主要用于收集查询条件和显示查询结果，并通过自定义的 JavaScript 函数验证输入的查询条件是否合法。该页面中所涉及的表单元素如表 24.5 所示。

表 24.5　图书借阅查询页面所涉及的表单元素

名　称	元素类型	重要属性	含　义
myform	form	method="post" action=""	表单
flag1	checkbox	value="a"	请选择查询依据
flag2	checkbox	value="b"	借阅时间
f	select	<option value="k.barcode" >图书条形码</option> <option value="k.bookname">图书名称</option> <option value="r.barcode">读者条形码</option> <option value="r.name">读者名称</option>	查询字段
key	text	size="50"	关键字
sdate	text	id="sdate"	开始日期
edate	text	id="edate"	结束日期
submit	submit	onClick="return check(myform);"	“查询”按钮

在图书借阅查询页面中指定查询条件后，提交表单信息到当前页。首先获取表单元素复选框 flag 的值，然后根据 flag 的值组合查询字符串。

如果 flag1 的值等于 a，那么按指定的字段检索图书借阅信息；如果 flag2 的值等于 b，那么按指定的时间段检索图书借阅信息；如果 flag1 的值等于 a，并且 flag2 的值等于 b，那么按以上两个条件的综合条件检索图书借阅信息，并将查询结果输出到浏览器。具体代码如下：

代码位置：光盘\mr\24\borrowQuery.php

```
<?php
include("conn/conn.php");                                          //连接数据库文件
$sql=mysql_query("select  b.borrowTime,b.backTime,b.ifback,r.barcode  as  readerbarcode,r.name,k.id,  k.barcode,
k.bookname from tb_borrow b join tb_reader r on b.readerid=r.id join tb_bookinfo k on b.bookid=k.id");
                                                                   //查询图书借阅信息
if($_POST[Submit]!=""){                                            //如果提交了表单，则执行以下操作
    $f=$_POST[f];                                                  //获取操作员选择的查询条件
    $key1=$_POST[key1];                                            //获取查询关键字
    $sdate=$_POST[sdate];                                          //获取借阅的起始日期
    $edate=$_POST[edate];                                          //获取借阅的结束日期
    $flag1=$_POST[flag1];                                          //获取按指定条件查询的复选框值
    $flag2=$_POST[flag2];                                          //获取按日期查询的复选框值
    if($flag1=="a"){                                               //如果按指定条件查询，则执行以下语句
    $sql=mysql_query("select b.borrowTime,b.backTime,b.ifback,r.barcode as
 readerbarcode,r.name,k.id,k.barcode,k.bookname  from  tb_borrow  b  join  tb_reader  r  on  b.readerid=r.id  join
tb_bookinfo k on
b.bookid=k.id where $f like '%$key1%'");
    }
    if($flag2=="b"){                                               //如果按时间段查询，则执行以下语句
    $sql=mysql_query("select b.borrowTime,b.backTime,b.ifback,r.barcode as
 readerbarcode,r.name,k.id,k.barcode,k.bookname  from  tb_borrow  b  join  tb_reader  r  on  b.readerid=r.id  join
tb_bookinfo k on
b.bookid=k.id where borrowTime between '$sdate' and '$edate'");
```

```
        }
        if($flag1=="a" && $flag2=="b"){                                    //如果按综合条件查询，则执行以下语句
        $sql=mysql_query("select b.borrowTime,b.backTime,b.ifback,r.barcode as
 readerbarcode,r.name,k.id,k.barcode,k.bookname from tb_borrow b join tb_reader r on b.readerid=r.id join
tb_bookinfo k on
b.bookid=k.id where borrowTime between '$sdate' and '$edate' and $f like '%$key1%'");
        }
}
$result=mysql_fetch_array($sql);                                           //检索查询结果
if($result==false){                                                        //如果查询结果不存在，则弹出提示信息
?>
<table width="100%" height="30"  border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td height="36" align="center">暂无图书借阅信息！</td>
  </tr>
</table>
<?php
}
else{                                                                      //否则，输出图书借阅信息
?>
<table width="723"  border="1" cellpadding="0" cellspacing="0" bordercolor="#FFFFFF" bordercolordark="#D2E3E6"
bordercolorlight="#FFFFFF">
  <tr align="center" bgcolor="#D0E9F8">
    <td width="13%">图书条形码</td>
    <td width="27%">图书名称</td>
    <td width="15%">读者条形码</td>
    <td width="11%">读者名称</td>
    <td width="13%">借阅时间</td>
    <td width="11%">归还时间</td>
    <td width="10%">是否归还</td>
  </tr>
<?php
do{
if($result[ifback]=="0"){                                                  //如果"是否归还"等于 0，则输出"未归还"
        $ifbackstr="未归还";
}
else{                                                                      //如果"是否归还"等于 1，则输出"已归还"
        $ifbackstr="已归还";
}
?>
/* *******************************输出符合查询条件的记录**************************** */
<tr>
     <td style="padding:5px;"> <?php echo $result[barcode];?></td>
     <td style="padding:5px;"><a href="book_look.php?id=<?php echo $result[id]; ?>"><?php echo $result
[bookname]; ?></a></td>
     <td style="padding:5px;"> <?php echo $result[readerbarcode];?></td>
     <td style="padding:5px;"> <?php echo $result[name];?></td>
     <td style="padding:5px;"> <?php echo $result[borrowTime];?></td>
     <td style="padding:5px;"> <?php echo $result[backTime];?></td>
     <td style="padding:5px;"> <?php echo $ifbackstr;?></td>
```

```
</tr>
/* ****************************************************************************************** */
<?php
        }while($result=mysql_fetch_array($sql));
}
?>
```

24.9 开发技巧与难点分析

24.9.1 如何自动计算图书归还日期

在图书馆管理系统中会遇到这样的问题：在借阅图书时，需要自动计算图书的归还日期。

1．图书归还日期

根据图书馆还书的规律一般都以 30 天为一个期限，因此在图书归还时，可以设置一个固定的值，即 30 天。计算归还日期的方法如下：

图书归还日期=“系统当前日期”＋“借阅天数固定值 30 天”。

自动计算图书归还日期的具体代码如下：

```
date("Y-m-d",(time()+3600*24*30))                    //图书归还日期
```

2．续借图书归还日期

续借图书归还日期是在原来数据库保存该图书归还日期（这个日期是不固定的）的基础上而再次借阅所计算的时间，它需要根据数据表中保存归还日期来计算。计算图书续借归还日期的方法如下：

续借图书归还日期=“所借图书在数据表中的归还日期”＋“借阅天数固定值 30 天”。

首先应用 substr()函数分别取出所借图书在数据表中原定的归还日期“月”、“日”、“年”，然后应用 mktime()函数计算出归还日期的时间戳，最后应用 date()函数格式化日期为“YYYY-MM-DD”格式。自动计算续借图书归还日期的代码如下：

```
$new=$_GET[backTime];                    //获取传递过来的该图书在数据表中的归还日期
//更新续借期，将动态获取的还书日期转化为时间截，然后再求出续借后的还书日期
date("Y-m-d",(mktime(0, 0, 0, substr($new,5,2), substr($new,8,2), substr($new,0,4))+3600*24*30));
```

24.9.2 如何对图书借阅信息进行统计排行

在图书馆管理系统的首页中，提供了显示图书借阅排行榜功能。要实现该功能，最重要的是如何获取统计排行信息，这可以通过一条 SQL 语句实现。本系统中实现对图书借阅信息进行统计排行的 SQL 语句如下：

```
select * from (select bookid,count(bookid) as degree from tb_borrow group by bookid) as borr join (select b.*,c.name as bookcasename,p.pubname,t.typename from tb_bookinfo b left join tb_bookcase c on b.bookcase=c.id join tb_publishing p on b.ISBN=p.ISBN join tb_booktype t on b.typeid=t.id where b.del=0) as book on borr.bookid=book.id order by borr.degree desc limit 10
```

下面对该 SQL 语句进行分析：

（1）对图书借阅信息表进行分组并统计每本图书的借阅次数，然后使用 as 为其指定别名为 borr。代码如下：



```
(select bookid,count(bookid) as degree from tb_borrow group by bookid) as borr
```

（2）使用左联接查询出图书的完整信息，然后使用 as 为其指定别名为 book。代码如下：

```
(select b.*,c.name as bookcasename,p.pubname,t.typename from tb_bookinfo b left join tb_bookcase c on b.bookcase=c.id join tb_publishing p on b.ISBN=p.ISBN join tb_booktype t on b.typeid=t.id where b.del=0) as book
```

（3）使用 join on 语句将 borr 和 book 连接起来，再对其按统计的借阅次数 degree 进行降序排序，并使用 limit 子句限制返回的行数。

24.10　开 发 总 结

本堂课运用软件工程的设计思想，通过一个完整的图书馆管理系统引导读者深入了解系统的开发流程。在这个系统的实现过程中，除了应用一些基本的 PHP 技术之外，还涉及一些独特的技术细节，如权限设置、多表查询技术等需要读者掌握，并能在实际的操作中灵活应用，举一反三。

附录 A　PHP 编码规则

PHP 编码规则不是一个定理、一个公式，需要 PHP 程序员去刻意的遵循，但是如果 PHP 程序员遵循 PHP 编码规则，那么不但可以提高代码的质量，还可以提高程序的可维护性、开发速度和效率；如果是团队开发项目，还可以让参与项目的其他开发人员更容易了解项目中的代码、弄清程序的状况，更快地投入到项目开发的工作中来。

如果是闭门造车，养成一套自己的编码风格，结果导致编写的东西只有自己可以看懂，其他人在阅读时浪费过多的时间和精力，不易于对程序的维护、更新。如果是团队开发项目，将很难投入到项目的开发工作中，因为其他人不了解你的编码规则，不能快速地阅读你的程序代码。

A.1　PHP File 文件格式

A.1.1　常规

对于只包含有 PHP 代码的文件，结束标志（"?>"）是不允许存在的，PHP 自身不需要（"?>"），这样做，可以防止它的末尾被意外地注入空白并显示输出。

由__HALT_COMPILER()允许的任意的二进制代码的内容被 Zend Framework PHP 文件或由它们产生的文件禁止。这个功能的使用只对特殊的安装脚本开放。

A.1.2　缩进

使用 4 个空格的缩进，而不使用制表符 Tab。

A.1.3　行的最大长度

一行 80 字符以内比较合适，长点也可以，但最多为 120 个字符。

A.1.4　行结束标志

行结束标志只能是标准的 UNIX 文本文件的换行，换行符在文件中表示为 10，或十六进制的 0x0A。

不要使用 Macintosh 电脑的回车（如 0x0D）。

不要使用 Windows 电脑的回车换行组合（如 0x0D、0x0A）。

A.1.5　引号

PHP 中单引号和双引号具有不同的含义，其主要区别如下：

（1）单引号中，任何变量（$var）、特殊转义字符（如“\t \r \n”等）不会被解析，因此 PHP 的解析速

度更快，转义字符仅仅支持“\'”和“\\”这样对单引号和反斜杠本身的转义。

（2）双引号中，变量（$var）值会代入字符串中，特殊转义字符也会被解析成特定的单个字符，还有一些专门针对上述两项特性的特殊转义，例如“\$”和“{$array['key']}”。这样虽然程序编写更加方便，但同时 PHP 的解析也很慢。

（3）数组中，如果下标不是整型，而是字符串类型，必须使用单引号将下标括起，正确的书写为$array['key']，而不是$array[key]，因为不正确的写法将导致 PHP 解析器认为 key 是一个常量，进而先判断常量是否存在，不存在时才以 key 作为下标传入表达式中，同时触发错误事件，产生一条 Notice 级错误。

因此，在绝大多数可以使用单引号的场合，禁止使用双引号。根据上述分析，可以或必须使用单引号的情况如下：

（1）字符串为固定值，不包含“\t”等特殊转义字符。

（2）数组的固定下标，如$array['key']。

（3）表达式中不需要传入变量，如$string = 'string'，而非$string = 'string$var'。

（4）还有一些特例，在正则表达式（preg_系列函数和 ereg 系列函数）中，全部使用双引号，这是为了人工分析和编写的方便，并保持正则表达式的统一，减少不必要的分析混淆。

（5）数据库 SQL 语句中，所有数据都不得加单引号，但是在进行 SQL 查询之前都必须经过 intval 函数处理；所有字符串都必须加单引号，以避免出现 SQL 注入漏洞。在 SQL 查询语句中的正确写法如下：

```
$user_id = intval($user_id);
select * form tb_user where user_name='$username' and user_id=$user_id;
```

（6）缓存文件中，一般对缓存数据的值采用 addcslashes($string, '\'\\')进行转义。

A.1.6　运算符、小括号、空格和关键词

每个运算符与两边参与运算的值或表达式中间要有一个空格，唯一的特例是字符连接运算符号两边不加空格。

左括号“(”应和函数关键词紧贴在一起，除此以外应当使用空格将“(”同前面内容分开。

右括号“)”除后面是“)”或者“.”以外，其他一律用空格分隔。

除字符串中有特殊情况外，在程序以及 HTML 中不出现两个连续的空格。

任何情况下，PHP 程序中不能出现空白的带有 Tab 或空格的行，即这类空白行应当不包含任何 Tab 或空格。同时，任何程序行尾也不能出现多余的 Tab 或空格。多数编辑器具有自动去除行尾空格的功能，如果习惯养成不好，可临时使用它，避免多余空格产生。

每段较大的程序体，上、下应当加入空白行，两个程序块之间只使用一个空行，禁止使用多行。

程序块划分尽量合理，过大或者过小的分割都会影响他人对代码的阅读和理解。一般以较大函数定义、逻辑结构、功能结构来进行划分。少于 15 行的程序块，可不加上下空白行。

说明或显示部分中，内容如含有中文、数字、英文单词混杂，应当在数字或者英文单词的前后加入空格。符合上述编码规范的示例代码如下：

```
$result = (($a + 1) * 3 / 2 + $num)).'Test';
$condition ? func1($var) : func2($var);
$condition ? $long_statement
: $another_long_statement;
if ($flag)
{
```



```
//Statements
//More than 15 lines
}
Showmessage(‘请使用 restore.php 工具恢复数据。’);
```

A.2　命名约定

A.2.1　类

类名只允许有字母、数字和字符，但不鼓励使用数字。下划线只允许做路径分隔符，例如，Zend/Db/Table.php 文件中对应的类名称是 Zend_Db_Table。

如果类名包含多个单词，每个单词的第一个字母必须大写，连续的大写是不允许的，例如，"Zend_PDF"是不允许的，而"Zend_Pdf"是可接受的。

由 Zend 或其参与 Zend Framework 项目的伙伴公司发行的类必须以"Zend_"开头并且必须按等级放在"Zend/"目录下。

A.2.2　接口

接口类也必须遵循同样的约定（如上所述），但必须以"Interface"结尾，例如：

```
Zend_Log_Adapter_Interface
Zend_Controller_Dispatcher_Interface
```

A.2.3　文件名

对于其他文件，只有字母、数字、字符、下划线和短横线（"-"）可用，空格是不允许的。

包含任何 PHP 代码的任何文件必须以".php" 扩展名结尾。

普通程序，能够被 URL 直接调用的程序，如 list.php、index.php，可以直接使用程序名+.php 的方式命名。

函数库和类库程序，应以小写.func.php 和.class.php 作为扩展名。函数库和类库程序只能被其他程序引用，而不能独立运行。其中不能包含任何流程性的、不属于任何函数或类的程序代码。

流程性程序，以小写.inc.php 作为扩展名。只能被其他程序引用，而不能独立运行。其中不能包含任何函数或类代码的程序代码。

模板源文件，以小写.html 作为扩展名。模板源文件按照 PHP 模板编码规则进行编写，不是可执行的程序，只能被 PHP 模板编译器所解析，放置于./templates/default 或./templates 下。

模板目标文件，模板文件被编译后自动生成的目标程序，以小写.php 作为扩展名，存放于./data/templates 目录下。

语言包文件，以小写.lang.php 作为扩展名，只能存放模板或程序使用的语言包信息。

缓存文件，此类文件为系统自动生成，以 cache_xxx.php、usergroup_xxx.php、style_xxx.php 等类似形式命名，存放于./data/cache 目录下。

文件名必须遵循上述的对应类名的规则。

A.2.4 函数和方法

函数名只包含字母、数字、字符，但不鼓励使用数字，下划线是不允许的。

函数名总是以小写开头，当函数名包含多个单词，每个词的首字母必须大写，这就是所谓的“驼峰”格式。

我们鼓励使用冗长的名字，这样容易理解代码。下面是符合约定的命名：

```
filterInput()
getElementById()
widgetFactory()
```

对于面向对象编程，对象的访问器总是以"get"或"set"为前缀。当使用设计模式如单态模式（singleton）或工厂模式（factory），方法的名字应当包含模式的名字，这样容易从名字识别设计模式。

在对象中的方法，声明为"private"或"protected"的，名称的首字符必须是一个单个的下划线，这是唯一的下划线在方法名字中的用法。声明为"public"的从不以下划线开头。

全局函数（"floating functions"）允许但不鼓励，建议把这类函数封装到静态类中。

以标准计算机英文为蓝本，尽量避免使用拼音或拼音英文混合的命名方式。

函数的名字和变量的命名规范一致。

函数定义中的左小括号，与函数名之间无空格。

开始的左大括号另起一行。

具有默认值的参数应该位于参数列表的后面。

函数定义与调用的参数之间加入一个空格。

尽量避免函数起始缩进位置与结束缩进位置不同。

符合上述书写规则的函数定义方法如下：

```
function authcode($string, $operation, $key = '')
{
if($flag)
{
//Statement
}
//函数体
}
```

A.2.5 变量

变量只包含数字、字母、字符，不鼓励使用数字，下划线不接受。

声明为"private"或"protected" 的类成员变量名必须以一个单个下划线开头，这是唯一的下划线在变量名中的用法，声明为"public"的从不以下划线开头。

与函数名一样，变量名总以小写字母开头并遵循“驼峰式”命名约定。

我们鼓励使用冗长的名字，这样容易理解代码。除非在小循环中，不鼓励使用简洁的名字如"$i"和"$n"。如果一个循环超过 20 行代码，索引的变量名必须是具有描述意义的名字。

变量命名只能使用项目中有据可查的英文缩写方式，例如可以使用$data 而不可使用$data1、$data2 这样容易产生混淆的形式，应当使用$username、$password 这样一目了然容易理解的形式。

可以合理地对过长的命名进行缩写，如$bio($biography)、$tpp($threadsPerPage)，前提是英文中有这样



既有的缩写形式，或字母符合英文缩写规范。

必须清楚所使用英文单词的词性，在权限相关的范围内，大多使用$enable***、$is***的形式，前者后面接动词，后者后面接形容词。

下面是一些基本的命名规则：

☑ 名称的开头不能使用数字及特殊符号。

☑ “·”及类型声明等专用语不能作为名称。

☑ 变量或程序名的长度必须在 255 个字符以内。

☑ 和保留字相同的名称不能使用。

A.2.6　常量

常量包含数字、字母、字符和下划线，数字允许作为常量名。

常量名的所有字母必须大写。

为加强可读性，常量中的单词必须以下划线分隔。例如，可以是这种格式： EMBED_SUPPRESS_EMBED_EXCEPTION，但是不允许使用这种格式：EMBED_SUPPRESSEMBEDEXCEPTION。

常量必须通过"const"定义为类的成员，不鼓励使用 "define" 定义的全局常量。

A.2.7　数据库表和字段

表和字段的命名以命名原则中定义的规范为依据。

所有数据表名称，只要其名称是可数名词，应尽量以复数方式命名；存储多项内容的字段，或代表数量的字段，也应当以复数方式命名，如 hits（查看次数）、items（内容数量）。

当几个表间的字段有关联时，要注意表与表之间关联字段命名的统一，如 forum_articles 表中的 articleid 与 forum_restores 表中的 articleid。

代表 id 自增量的字段，通常用以下几种形式：

☑ 一般情况下，使用全称的形式，如 userid、articleid。

☑ 没有功能性作用，只为管理和维护方便而设的 id，可以使用全称的形式，也可只将其命名为 id。

A.3　编 码 风 格

A.3.1　PHP 代码划分（Demarcation）

PHP 代码总是用完整的、标准的 PHP 标签定界，例如：

```
<?php

?>
```

短标签（<? ?>）是不允许的，只包含 PHP 代码的文件，不要结束标签（参见 A.1.1 节）。

A.3.2　字符串

1．字符串文字

当字符串是文字（不包含变量）时，用单引号（apostrophe）括起来，例如：

```
$a = 'Example String';
```

2．包含单引号（'）的字符串文字

当文字字符串包含单引号（apostrophe）就用双引号括起来，特别在 SQL 语句中：

```
$sql = "SELECT `id`, `name` from `people` WHERE `name`='Fred' OR `name`='Susan'";
```

在转义单引号时，上述语法是首选。

3．变量替换

变量替换有下面两种形式：

```
$greeting = "Hello $name, welcome back!";
$greeting = "Hello {$name}, welcome back!";
```

为保持一致，下面的形式是不允许的：

```
$greeting = "Hello ${name}, welcome back!";
```

4．字符串连接

字符串用 "." 操作符连接，在它的前后加上空格以提高可读性，例如：

```
$project = 'PHP' . ' ' . 'Zend';
```

当用 "." 操作符连接字符串，代码可以分成多个行，也是为提高可读性。在这些例子中，每个连续的行应当由 whitespace 来填补，例如 "." 和 "=" 对齐：

```
$sql = "SELECT `id`, `name` FROM `tb_admin ` "
     . "WHERE `name` = 'Susan' "
     . "ORDER BY `name` ASC ";
```

A.3.3　数组

1．数字索引数组

索引不能为负数。

建议数组索引从 0 开始。

当用 array 声明有索引的数组，在每个逗号的后面间隔空格以提高可读性，例如：

```
$exampleArray = array(a, b, c, 'PHP', 'Studio');
```

也可以用 "array" 声明多行有索引的数组，在每个连续行的开头要用空格填补对齐，例如：

```
$exampleArray = array(a, b, c, 'PHP', 'Studio',
                      $a, $b, $c,
                      56.44, $d, 500);
```

2．关联数组

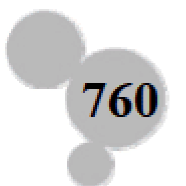
当用 array 声明关联数组时，鼓励把代码分成多行，在每个连续行的开头用空格填补来对齐键和值，例如：

```
$exampleArray = array('firstKey'   => 'firstValue',
                      'secondKey' => 'secondValue');
```

A.3.4　类

1．类的声明

用下面的约定来命名类。



☑　花括号总是从类名下一行开始。
☑　每个类必须有一个符合 PHP 编码标准的文档块。
☑　4 个空格的缩进。
☑　每个 PHP 文件中只有一个类。
☑　在类中允许放置另外的代码，但不鼓励这样做。在这些文件中，通过两行空格来分隔类和其他代码。示例如下：

```
/**
 * Documentation Block Here
 */
class ExampleClass
{
    // entire content of class
    // must be indented four spaces
}
```

2．类成员变量

必须用下面的变量名约定来命名类成员变量。
☑　变量的声明必须在类的顶部，要先于方法的声明。
☑　不允许使用 var，要使用 private、protected 或 public。

A.3.5　函数和方法

1．函数和方法声明

必须用下面的变量名约定来命名函数。
☑　在类中的函数必须用 private、protected 或 public 声明它们的可见性。
☑　像类一样，花括号从函数名的下一行开始，函数名与包括参数的圆括号中间没有空格。
☑　反对使用全局函数。
示例如下：

```
/**
 * Documentation Block Here
 */
class Euspice
{
    /**
     * Documentation Block Here
     */
    public function cattle()
    {
        // entire content of function
        // must be indented four spaces
    }
}
```

2．函数和方法的用法

函数的参数用逗号和空格将紧接着的参数分隔，下面例子中的函数带有 3 个参数：

```
threeArguments(1, 2, 3);
```

传值方式在调用时是禁止的。

带有数组参数的函数，函数的调用可包括 array 并分成多行来提高可读性，同时，书写数组的标准仍然适用，例如：

```
threeArguments(array(1, 2, 3), 2, 3);
threeArguments(array(1, 2, 3, 'Zend', 'Studio',
                     $a, $b, $c,
                     56.44, $d, 500), 2, 3);
```

A.3.6 控制语句

1．if/else/elseif

使用 if/else/elseif 的控制语句，在条件语句的圆括号前后都必须有一个空格。

在圆括号里的条件语句，操作符必须用空格分开，鼓励使用多重圆括号以提高在复杂的条件中划分逻辑组合。

前花括号必须和条件语句在同一行，后花括号单独在最后一行，其中的内容用 4 个空格缩进。

```
if ($a != 2) {
    $a = 2;
}
```

下面的例子应用 if 语句，包括 elseif 或 else 的格式约定：

```
if ($a != 2) {
    $a = 2;
} else {
   $a = 7;
}
if ($a != 2) {
    $a = 2;
} elseif ($a == 3) {
   $a = 4;
} else {
   $a = 7;
}
```

在有些情况下，PHP 允许这些语句不用花括号，但在 ZF 代码标准中，它们（if、elseif 或 else 语句）必须使用花括号。

elseif 是允许的但不鼓励，我们支持 else if 组合。

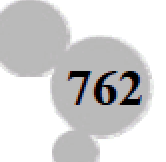
2．switch

在 switch 结构的控制语句中，在条件语句的圆括号前后必须都有一个单个的空格。

switch 中的代码必须有 4 个空格缩进，在 case 中的代码再缩进 4 个空格。

```
switch ($numPeople) {
    case 1:
        break;

    case 2:
        break;
```



```
    default:
        break;
}
```

switch 语句中必须有 default。

A.3.7 数据库

1. 字段结构

允许 null 值的字段，数据库在进行比较操作时，会先判断其是否为 null，非 null 时才进行值的比对。因此基于效率的考虑，所有字段均不能为空，即全部 not null。

2. SQL 语句

所有 SQL 语句中，除了表名、字段名称以外，全部语句和函数均需大写，应当杜绝小写方式或大小写混杂的写法。例如，select * from tb_user;是不符合规范的写法。

长 SQL 语句应当有适当的断行，依据 JOIN、FROM、ORDER BY 等关键字进行界定。

通常情况下，在对多表进行操作时，要根据不同表名称，对每个表指定一个 1～2 个字母的缩写，以利于语句简洁性和可读性。下面的 SQL 语句是符合编码规范的：

```
$result = $db->query("Select m.*, i.*
FROM ".TABLE_MEMBER." m, ".TABLE_MEMBERINFO." i
Where m.userid=i.userid AND m.userid='$_userid'");
```

3. 定长与变长表

包含任何 varchar、text 等变长字段的数据表，即为变长表，反之则为定长表。

对于变长表，由于记录大小不同，在其上进行许多删除和更改将会使表中的碎片更多。需要定期运行 OPTIMIZE TABLE 以保持性能，而定长表就没有这个问题。

如果表中有可变长的字段，将它们转换为定长字段能够改进性能，因为定长记录易于处理。但在试图这样做之前，应该考虑下列问题：

（1）使用定长列涉及某种折中。它们更快，但占用的空间更多。char(n)类型列的每个值总要占用 n 个字节（即使空串也是如此），因为在表中存储时，值的长度不够将在右边补空格。

（2）而 varchar(n)类型的列所占空间较少，因为只给它们分配存储每个值所需要的空间，每个值再加一个字节用于记录其长度。因此，如果在 char 和 varchar 类型之间进行选择，需要对时间与空间作出折中。

变长表到定长表的转换，不能只转换一个可变长字段，必须对它们全部进行转换。而且必须使用一个 Alter TABLE 语句同时全部转换，否则转换将不起作用。

有时不能使用定长类型，即使想这样做也不行。例如，对于比 255 字符更长的串，没有定长类型。

在设计表结构时如果能够使用定长数据类型尽量用定长的，因为定长表的查询、检索、更新速度都很快。必要时可以把部分关键的、承担频繁访问的表拆分。

进行表结构设计时，应当做到恰到好处，反复推敲，从而实现最优的数据存储体系。

4. 结构优化与索引优化

索引能加快查询速度，而索引优化和查询优化是相辅相成的，既可以依据查询对索引进行优化，也可以依据现有索引对查询进行优化，这取决于修改查询或索引。

索引优化与查询优化的最基本准则介绍如下。

（1）根据产品的实际运行和被访问情况，找出哪些 SQL 语句是最常被执行的。最常被执行和最常出现在程序中是完全不同的概念。最常被执行的 SQL 语句，又可被划分为对大表（数据条目多的）和对小表（数据条目少的）的操作。无论大表或小表，又可分为读（Select）多、写（Update/Insert）多或者读写都多的操作。

对常被执行的 SQL 语句而言，对大表操作需要注意：

写操作多的，通常可使用写入缓存的方法，先将需要写或需要更新的数据缓存至文件或其他表，定期对大表进行批量写操作。同时，应尽量使得常被读写的大表为定长类型，即便原本的结构中大表并非定长。大表定长化，可以通过改变数据存储结构和数据读取方式，将一个大表拆成一个读写多的定长表，和一个读多写少的变长表来实现。

读操作多的，需要依据 SQL 查询频率设置专门针对高频 SQL 语句的索引和联合索引。

而小表就相对简单，加入符合查询要求的特定索引，通常效果比较明显。同时，定长化小表也有益于效率和负载能力的提高。字段比较少的小定长表，甚至可以不需要索引。

（2）看 SQL 语句的条件和排序字段是否动态性很高（即根据不同功能开关或属性，SQL 查询条件和排序字段的变化很大的情况），动态性过高的 SQL 语句是无法通过索引进行优化的。唯一的办法只有将数据缓存起来，定期更新，适用于结果对实效性要求不高的操作。

事实上，索引是将条件查询、排序的读操作的资源消耗，分布到了写操作中，索引越多，耗费磁盘空间越大，写操作越慢。因此，索引决不能盲目添加。对字段索引与否，最根本的依据是 SQL 语句执行的概率、表的大小和写操作的频繁程度。

A.3.8 注释文档

程序注释中要用简单、明了、精确的语言表达出程序的处理说明。避免每行程序都使用注释，可以在一段程序的前面加一段注释，具有明确的处理逻辑。注释虽然必不可少，但也不应过多，不要被动地为写注释而写注释。

1. 文件

每个包含 PHP 代码的文件必须至少在文件顶部包含 phpDocumentor 标签，例如：

```
/**
 * 文件的简短描述
 *
 * 文件的详细描述（如果有的话）... ...
 *
 * LICENSE: 一些 license 信息
 *
 * @copyright   2005 Zend Technologies
 * @license     http://www.zend.com/license/3_0.txt     PHP License 3.0
 * @version     $Id:$
 * @link        http://dev.zend.com/package/PackageName
 * @since       File available since Release 1.2.0
*/
```

2. 类

每个类必须至少包含 phpDocumentor 标签，例如：



```
/**
 * 类的简述
 *
 * 类的详细描述 （如果有的话）... ...
 *
 * @copyright   2005 Zend Technologies
 * @license      http://www.zend.com/license/3_0.txt     PHP License 3.0
 * @version      Release: @package_version@
 * @link          http://dev.zend.com/package/PackageName
 * @since         Class available since Release 1.2.0
 * @deprecated Class deprecated in Release 2.0.0
 */
```

3. 函数

每个函数，包括对象方法，必须有最少包含下列内容的文档块（docblock）：

☑ 函数的描述。

☑ 所有参数。

☑ 所有可能的返回值。

☑ 因为访问级已经通过"public"、"private"或"protected"声明，不需要使用"@access"。

☑ 如果函数/方法抛出一个异常，使用@throws，例如：

```
@throws exceptionclass [description]
```

4. 块注释

块注释通常用于提供对文件、方法、数据结构和算法的描述。块注释被置于每个文件的开始处以及每个方法之前。它们也可以被用于其他地方，如方法内部。在功能和方法内部的块注释应该和它们所描述的代码具有一样的缩进格式。

块注释之首应该有一个空行，用于把块注释和代码分割开来，例如：

```
/*
            * 这里是块注释
*/
```

5. 单行注释

短注释可以显示在一行内，并与其后的代码具有一样的缩进层级。如果一个注释不能在一行内写完，就该采用块注释。单行注释之前应该有一个空行。以下是一个代码中单行注释的例子：

```
If (condition) {
      /* 以下代码运行的条件 */
...
}
```

6. 尾端注释

极短的注释可以与它们所要描述的代码位于同一行，但是应该有足够的空白来分开代码和注释。若有多个短注释出现于大段代码中，它们应该具有相同的缩进。

以下是一个代码中尾端注释的例子：

```
If ($a == 2) {
return TRUE; /* 对单一条件的说明 */
} Else {
```

```
return isPrime($a); /*  其余的条件  */
}
```

7．行末注释

注释界定符“//”，可以注释掉整行或者一行中的一部分。

```
If ($Euspice > 1) {
    ...      //注释用法
}
```

附录　自我测试参考答案

第 1 堂课　PHP 概述

一、选择题答案

1．BC
2．A
3．B
4．A
5．B

二、填空题答案

1．default_timezone_set
2．windows
3．http://127.0.0.1、http://localhost
4．4、MySQL
5．date

第 2 堂课　PHP 基础

一、选择题答案

1．C
2．ABCD
3．A
4．B
5．B

二、填空题答案

1．$b=$a;
2．.
3．4
4．0

5．define()

第 3 堂课　PHP 函数

一、选择题答案

1．A
2．A
3．A
4．B
5．D

二、填空题答案

1．echo、var_dump
2．rmdir
3．date
4．输出变量的相关信息
5．没有任何内容输出

第 4 堂课　PHP 流程控制语句

一、选择题答案

1．B
2．E
3．B
4．A
5．D

二、填空题答案

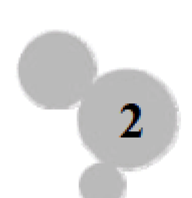
1．require()语句会输出错误信息，并且立即终止脚本的处理。而 include()语句在没有找到文件时则会输出警告，不会终止脚本的处理、require_once 或者 include_once。

2．面对对象程序设计、面向过程程序设计

3．sum=15

4．sum=55

5．会看到。



第 5 堂课　PHP 数组应用

一、选择题答案

1．B
2．A
3．C
4．D
5．D

二、填空题答案

1．size
2．array_push
3．array_unique()
4．数字索引数组
5．一维数组、二维数组、多维数组

第 6 堂课　PHP 与 Web 页面交互

一、选择题答案

1．D
2．C
3．A
4．D
5．B

二、填空题答案

1．$_POST[]、$_GET[]
2．onClick=" "
3．$_GET[]
4．函数名、参数、函数体、返回值
5．var

第 7 堂课　MySQL 数据库技术

一、选择题答案

1．D
2．D
3．A
4．D
5．A

二、填空题答案

1．delete from tb_user where id=5
2．mysql_query("set names utf8");
3．MYSQLDUMP 命令、MYSQL 命令
4．select * from tb_mrbook order by price desc limit 3;
5．0、1、2、3、4、5 条记录的所有字段信息

第 8 堂课　PHP 数据库编程技术

一、选择题答案

1．A
2．D
3．A
4．B
5．A

二、填空题答案

1．name='".$_POST['name']."'
2．

```
$conn=    mysql_connect    ("localhost","root","111");
mysql_select_db    ("db_database08",   $conn   );
mysql_query    ("set names utf8");
```

3．mysql_fetch_array()和 mysql_fetch_row()
4．id<5
5．'mr','123456'



第 9 堂课　Cookie 与会话控制

一、选择题答案

1．B
2．A
3．A
4．C
5．C

二、填空题答案

1．isset($_SESSION['mr'])
2．setCookie("mr",'明日科技');
3．$_COOKIE['mr']
4．session_destroy();
5．空

第 10 堂课　字符串高级处理技术

一、选择题答案

1．A
2．A
3．B
4．A
5．B

二、填空题答案

1．Explode、implode、print
2．This,course,is,very,easy!
3．POSIX　PCRE
4．strlen
5．preg_match

第 11 堂课　日期和时间的处理技术

一、选择题答案

1. C
2. B
3. C
4. C
5. B

二、填空题答案

1. Strtotime("1 January 2100")
2. Date_default_timezone_set("Asia/Shanghai");
3. mktime(date("Y-m-d"));
4. date_default_timezone_set("Asia/Shanghai");

```
$str=mktime("2010-12-10");
 $str=getdate($str);
 echo $str[weekday];
```

5. 今天不是一个特殊的日子！

第 12 堂课　图形图像处理技术

一、选择题答案

1. C
2. B
3. A
4. C
5. C

二、填空题答案

1. $image=imagecreate(800,800);
2. imagedestroy;
3. imagecreate;
4. imagestring;
5. 130、30



第 13 堂课　文件目录处理技术

一、选择题答案

1．C
2．C
3．A
4．B
5．C

二、填空题答案

1．$array=$_FILES['file']
2．rmdir
3．Move_uploaded_file
4．Is_uploaded_file
5．下载

第 14 堂课　面向对象的程序设计

一、选择题答案

1．C
2．A
3．C
4．C
5．B

二、填空题答案

1．Abstruct class People{}
2．clone
3．static
4．instanceof
5．我是初始化函数，在实例化的同时就被调用了

第 15 堂课　PHP 与 XML 技术

一、选择题答案

1．C
2．A
3．B
4．A
5．C

二、填空题答案

1．$dom = new DomDocument('1.0','gb2312');
2．Simplexml_load_file()
3．Simplexml_load_string()
4．Simplexml_load_date()
5．createTextNode()

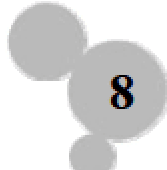
第 16 堂课　PDO 数据库抽象层

一、选择题答案

1．B
2．A
3．A
4．C
5．B

二、填空题答案

1．PDO::ERRMODE_SILENT
2．PDO::ERRMODE_WARNING
3．PDO::ERRMODE_EXCEPTION
4．Exec
5．PDO 连接 MySQL 成功



第 17 堂课　Smarty 模板技术

一、选择题答案

1．A
2．A
3．D
4．D
5．D

二、填空题答案

1．templates
2．$smarty->caching=true
3．display
4．assign
5．$smarty->assign("value",$rs);

第 18 堂课　Zend Framework 框架

一、选择题答案

1．A
2．C
3．A
4．A
5．A

二、填空题答案

1．Zend_Db
2．Zend_Files
3．Zend_Layout
4．Zend_Mail
5．Zend_From

第 19 堂课　PHP 的安全与防护

一、选择题答案

1．C
2．C
3．B
4．A
5．A

二、填空题答案

1．crypt($string)
2．md5($string)
3．sha1($string)
4．限制文件上传的格式
5．将权限判断写入 php 代码中

第 20 堂课　PHP 的字符编码

一、选择题答案

1．A
2．B
3．C
4．A
5．C

二、填空题答案

1．<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
2．mb_detect_encoding()
3．header("Content-type:text/html;charset=utf-8");
4．mb_substr()
5．iconv("utf-8","gbk","PHP");

第 21 堂课　程序调试与错误处理

一、选择题答案

1．A
2．B
3．A
4．C
5．B

二、填空题答案

1．die("false");
2．@
3．$conn=mysql_connect("localhost","root","111");mysql_select_db ("db_database21",$conn);
4．allow_url_fopen=On
5．echo mysql_error()